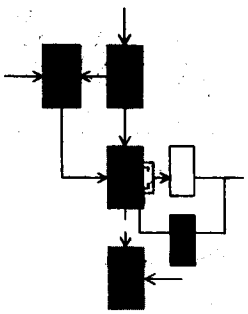
MAY, 1967

REPORT ESL-R-307
M.I.T. PROJECT DSR 76265
NASA Research Grant NGR-22-009(124)



COMPUTATION OF APPROXIMATE FUEL-OPTIMAL CONTROL

Donald L. Gray

Electronic Systems Laboratory

MASSACHUSETTS INSTITUTE OF TECHNOLOGY, CAMBRIDGE, MASSACHUSETTS 02139

Department of Electrical Engineering

May 1967 Report ESL-R-307 Copy No. 35

COMPUTATION OF APPROXIMATE FUEL-OPTIMAL CONTROL

by

Donald Lee Gray

This study was conducted at the Electronic Systems Laboratory with support extended in part by the National Aeronautics and Space Administration under Research Grant NsG-496 with the Center for Space Research, and in part by the National Aeronautics and Space Administration under Research Grant Number NGR-22-009(124) (M.I.T. Project Number DSR 76265). Reproduction in whole or in part is permitted for any purpose of the United States Government.

Electronic Systems Laboratory
Department of Electrical Engineering
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

ABSTRACT

An iterative digital computer method for determining the optimal control function is developed and tested. The class of problem treated is fixed time fuel-optimal control of a linear time-invariant plant to a given point. A sequence of suboptimal controls is produced each of which is efficient in use of fuel and does not require the fast switching time of the optimal control. Convergence of the method is proven under suitable assumptions. A Fortran program is given and computer results are presented for a number of examples. These examples illustrate the usefulness of the method. Ways of extending the method to other classes of problems are outlined.

ACKNOWLEDGEMENT

The material presented in this report is based on a thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at the Massachusetts Institute of Technology, May, 1967.

The author wishes to express his gratitude to Professor Michael Athans for his supervision and encouragement, his many helpful comments and suggestions, his well organized teaching, and his willingness to share an extensive collection of reprints. Special thanks are due to Professor Henry M. Paynter, who acted as chairman of the committee and whose lucid view of technology has greatly influenced the author. Professor Leonard Gould and Professor Roger Brockett acted as readers, gave helpful criticisms, and also offered advice and encouragement during the research.

The teaching and comments of Professors Elijah Polak (visiting from the University of California at Berkeley), Arthur E. Bryson Jr. (of Harvard University), W. Gilbert Strang (of the Department of Mathematics), and Fred C. Schweppe have been especially helpful.

Thanks are owed to the personnel at the M.I.T. Computation Center for their assistance in obtaining the computational results of this research.

The author is also grateful to the Publications Office and the Drafting Section of the M.I.T. Electronic Systems Laboratory for typing the thesis and preparing the figures.

CONTENTS

CONTENTS (Contd.)

CONTENTS (Contd.)

LIST OF FIGURES

CHAPTER I
INTRODUCTION

In this chapter the motivation and overall idea of the thesis are given. The purpose is to show how this computational method relates to previous work in the field. The design aims used are specified, and it is shown how the computational method was derived. Some conclusions are drawn from the experimental results. Finally, the author's opinion of the contributions of the thesis is stated.

The problem examined is fixed time fuel optimal control of a linear time invariant plant to a given state. Minimizing the total fuel used is of significance in many problems. One example is space flight in which the quantity of propellant used must be kept low. Another example is a chemical process in which the fuel could be the quantity of a certain chemical used or the number of kilowatt hours of power consumed.

For a brief treatment of the thesis work, it might be well to read Chapters I, II, and VI first. These chapters give the principal problem treated and the main idea. Chapter III then shows the important theoretical results and Chapters IV and V describe the computer work. Finally, Chapter VII outlines how a more difficult problem is attacked, and Chapter VIII indicates a number of extensions of the work.

A. BACKGROUND

Man has an innate desire to do things in a better way, or hopefully, in the best possible way. So the history of optimization methods is a very long one, with roots lost in antiquity. Instead of a listing of ancient work on optimization, just three landmarks will be mentioned as having direct bearing on the present study. The first is the invention of the (ordinary) calculus in the Seventeenth Century by Newton and Leibnitz. Besides solving a great many problems, the calculus led to a very basic necessary condition for any extreme value of an analytic function. This condition (the vanishing of the first derivative) is still in use in many forms today. The second landmark is the use of modern, high-speed computers to obtain numerical solutions. With this tool one can attack problems too complex or too long for reasonable hand calculations.

Since Newton's method will be used in what follows it seems appropriate to add a historical note about it here. The first recorded papers available today are those by Newton.[75,76] He applied the method

to finding the roots of polynomials in one or more dimensions, and included a primitive proof of conditions for convergence of the method.[75] Even Newton stated that the method might already be in use,* but he is credited as its founder. Moore, in Reference 2, says the method has a long history with contributions by Cauchy, Runge, Faber, and Blutel. For a summary of the modern method, see Appendix B.

Newton's method is used as a part of several computational methods for finding optimal controls, such as those of McReynolds and Bryson,[68] Knudsen,[58] Plant,[82] and Kleinman.[56] As a computational scheme in itself, Newton's method has been widely talked about and applied recently.[12, 16, 17, 28, 30, 42, 60, 66, 69, 89, 94] The main reason seems to be its rapid convergence. It is usually a surprise to someone used to slower methods to watch Newton's method zip right in to the answer: First one significant digit becomes correct, the next iteration two are correct, the next iteration four are correct, and the next iteration is as close as you can get with a 30-bit computer word length.

B. SUMMARY OF ITERATIVE METHODS

From this point on it is assumed that the optimization problem can be stated in a mathematical format such as that of Problem 1 of Chapter II or Problem 2 of Chapter VII. The performance criterion J is assumed to be additive. Then there are a number of approaches used for finding numerical solutions. Some of these are listed and/or described briefly below, starting with the most direct methods and proceeding to some more indirect methods. This is just a summary: much more complete lists can be found for instance in References 77, 79, 5, and 4.

1. The most straightforward way to compute an optimum solution is to search through the entire set of possible solutions, comparing the resulting values of performance to find the best one. In practice the set is scanned by using a discrete net of solutions. This assumes that two solutions "close" to each other in some valid distance function (see Appendix A) will also be close to each other in performance. The dis-

---

* "27. Whether this method of resolving equations be vulgarly practised I cannot tell, but surely to me it appears simple in comparison of others, and more accommodated to practice."

crete net of solution points to be evaluated can be chosen by either deterministic or random means. There are many recent references on the random or Monte Carlo method. For a simple introduction see e.g., Froberg.[31]

2. A more sophisticated approach to the searching technique is to use the given performance criterion J in some way to decide which solutions need to be examined. Techniques of linear and nonlinear programming fall into this category. For a simple introduction to programming methods see e.g., Part 3 of the symposium edited by Bellman.[15] Various ways of tearing the system into smaller parts, optimizing the smaller parts by multilevel programming, and then reconnecting have been avocated. For a sampling of the current literature on multilevel programming see for example Session VII of the 1965 Joint Automatic Control Conference.

Bellman's dynamic programming[99] is a logical way to eliminate many nonoptimal solutions on the basis of a partial computation. As with other search techniques, the large computer memory required for problems with more than three or four state variables is the chief limitation.

The gradient method suggested by Bryson and Denham,[20,24] and by Kelley[53,54] linearizes the problem about the most recent iteration and then seeks to change the control in the direction of maximum decrease of the penalty function. In order to do this the adjoint equations to the linearized differential equations are used to find the influence of a change in the control on the penalty function. This approach is simple to program, but suffers in many applications from a slow rate of convergence. A step size must also be chosen. More recently some work has been done on putting this method in the more general framework of nonlinear programming known as convex programming.[35,100]

The second variation method is suggested by members of the same research groups who suggest the gradient method.[19,52] By utilizing the second-order terms in a series expansion about the present iteration, the convergence of the method is greatly accelerated at the expense of greater complexity. If the expansion is about an optimal tragectory these second-order terms also lead to a method of feedback

control.[19] McReynolds and Bryson[68] suggest using a Riccati transformation and solving by a method of sweeps. This is equivalent to Newton's method.

3. Since Pontryagin's maximum principle[84] has become well known, a number of computational methods have been proposed which use information given by it. One interesting approach is to minimize (maximize) a sample Hamiltonian at each iteration, suggested by Kelley[53] and used by Gottlieb.[36] The first order change in the Hamiltonian has been used by Denn[102] and by Kurihari[103]. Durbeck[26] changes the performance index J in such a way as to minimize the sample Hamiltonian, which leads in the limit to efficient suboptimal controls.

Al'brekht[1] solves the Hamilton-Jacobi equation by a power series method, truncating to obtain a suboptimal control.

Application of the maximum principle leads to a two-point boundary value problem* (see e.g., Chapter II). A whole class of computational methods centers around solving the resulting TPBVP by finding the initial costate vector $\underline{\pi}^*$. This thesis develops a method of this class.

A fundamental property of the costate initial condition vector $\underline{\pi}^*$ is that it points opposite to the direction of the gradient of the minimum cost surface, whenever this gradient exists. (This implies that $\underline{\xi}'\underline{\pi}^* > 0$ under certain conditions, and that $\underline{\pi}^*$ can be searched for among the vectors $\underline{\pi}$ satisfying this inequality.)** Neustadt[74,73] has designed a computational method based on this property. Eaton[27] has extended the method to the rendezvous problem. Fadden and Gilbert in Reference 10 and Paiewonsky in Reference 30 have dealt with some computer aspects of this method.

Another method for finding time optimal controls is attributed to Krasovski and Gamkrelidze and discussed by Paiewonsky.[78] It involves choosing a final time T such that $T < T^*$ and finding a costate vector $\underline{\pi}$ such that the resulting initial state vector $\underline{x}(0)$ is colinear with the given initial condition $\underline{\xi}$. This is repeated by increasing T until $\underline{x}(0) \approx \underline{\xi}$.

---

*The two-point boundary value problem is abbreviated to TPBVP.

**See the Nomenclature Sheet for the meaning of these symbols.

The most direct method of finding the correct costate initial condition vector $\underline{\pi}^*$ is to guess a vector $\underline{\pi}$, solve the resulting initial value problem to obtain the terminal errors, and then correct $\underline{\pi}$ iteratively to decrease the errors. Bass[13] has applied this to a class of nonlinear state equations. Paiewonski[78] uses linear state equations and a gradient method for correcting $\underline{\pi}$. Knudsen[58] uses linear state equations and a form of Newton's method for correcting $\underline{\pi}$ that leads to faster convergence. However, this approach does not always work because the first order effect of variations in $\underline{\pi}$ can be zero (on a switch curve in the state space).

Plant[81, 82, 83] modified the boundary conditions of the problem, replacing the given terminal state $\underline{\theta}$ by a hypersphere around $\underline{\theta}$. Then Knudsen's method was applied. This procedure got rid of the switch curves and hence eliminated the main difficulty with Knudsen's method. Plant states that the hypersphere can be made insignificantly small without affecting the iterative procedure appreciably in the cases studied.

Comment 1.1

One of the requirements in computation is to insure that the global optimal control has been found. There are two basic ways to do this. One is to adequately search the entire space of possible solutions for the optimal one. This leads to relatively simple schemes which tend to require large computer memories and large amounts of computation time.

The other way of insuring a global optimum is to find only the solutions which satisfy a set of necessary conditions for an optimum (i.e., Pontryagin's maximum principle) and then:

1. compare all the resulting extremal solutions to find the global optimum, or
2. appeal to the physical or engineering reasonableness of the solution to rule out any better solutions, or
3. show the uniqueness of the extremal solution.

C. DESIGN OF THE COMPUTATIONAL METHOD

A certain point of view should be kept in mind when considering iterative methods: Every iterative method involves replacing the optimal control problem in some manner by a sequence of simpler problems which converges (hopefully) to the given optimal control problem.

The idea is to try to improve on current iterative schemes. Using some information from Pontryagin's maximum principle can lead to a faster method, (if the resulting information is used to advantage), and in fact it was decided to attack the TPBVP. The author was attracted by Newton's method because of its rapid convergence and because of the sufficient conditions for convergence given by Kantorovich.[49]

Also the design approach used by Plant[82] was intriguing. He changed the terminal boundary condition from a point to a hypersphere, which smoothed the problem out and removed a difficulty found in Knudsen's method.[58] This was justified not only on the ground that it made the computational method work better, but also on the ground that for many engineering problems it was more sensible.

With all this in mind, a computational method was sought that would solve the TPBVP, use Newton's method, and somehow smooth out the TPBVP. Of course, Newton's method could be applied to the TPBVP directly, but there is no guarantee it would converge. In talks with Professor Athans two points became clear: (1) If the optimal control function were analytic instead of a discontinuous function then the convergence theorem of Kantorovich could be applied to it, and (2) If the TPBVP were linear Newton's method would by definition converge in one step. So it would seem logical to replace the nonlinear TPBVP by a sequence of smoother ones, starting with a linear TPBVP and converging to the given nonlinear one. Newton's method is used to find the solution of each member of the sequence in turn, leading to a sequence of suboptimal controls.

As soon as the basic idea was suggested some of the properties proved under proper assumptions in Chapter III began to become apparent. The sequence could come as close as desired to the original TPBVP while retaining a smoothness property. There exists such a sequence for which Newton's method will converge when applied to each smoothed TPBVP sequentially.

In order to show these properties and also to do some computer examples it was decided to concentrate on a relatively simple problem; that of fuel-optimal, fixed-time control to a given state, with a linear time invariant plant (Problem 1). A mathematical development of the method for Problem 1 is shown in Chapter II, which is the key chapter in understanding the main part of the work.

D. EXPERIMENTAL RESULTS

A digital computer program was formulated, in Fortran II, based on the computational method designed above, and a number of numerical examples were run on an IBM 7094. In most cases the suboptimal controls were found and they converged to the optimal control. When this was not the case, usually the problem had no solution because the final state was outside the set of reachable states, or possible it was inside but very close to the edge of the set of reachable states.

One very important factor was the effect of numerical accuracy on the success of the method. For example, in a given problem the effect of varying the fineness of the mesh or partitioning used in approximate integration was studied. With a fine mesh the method works well; as the mesh is made more coarse, more iterations of Newton's method are needed, more members of the sequence of approximate TPBVP's are needed, and finally for some very low accuracy the method ceases to work at all. A conclusion reached by the author is that it is necessary to include information on mesh size, integration scheme, computer approximations and short cuts, etc., when describing results of computer studies with an iterative scheme.

The suboptimal controls were found to be very efficient in total fuel consumed. They transfer the system to the desired final state in the given time. Five or six members of the sequence of approximate operators were usually enough to come quite close to the exact TPBVP in terms of total fuel used ($< 1\%$ difference), appearance of the control function and the state space trajectory. From two to seven iterations of Newton's method were usually enough to solve a particular member of the sequence.

Reliability and generality were aimed for in the computer program more than fast computation time. Typical runs required from 8 to 100 seconds of elapsed computer time. It is difficult to make a direct comparision, but this computer program seems slower than that of Plant[82] and roughly comparable to or faster than some others.[60]

It was hoped that the convergence theorem of Kantorovich could be used to estimate when Newton's method would converge and when it would not. However, the computer examples showed it to be much too conservative as a sufficient condition for the class of problems studied. A sequence of approximate operators (TPBVP's) could be constructed so that the sufficient condition for convergence would be satisfied at every step, but it would require very many members for that sequence to converge to the exact operator. It was found better to use a step size based on a numerical study of the actual region of convergence for Newton's method with this class of problems. If convergence fails to occur for some operator, it is then easy to make a smaller change from the previous operator and try Newton's method again.

E. CONTRIBUTIONS OF THE THESIS

The main contribution of the thesis is the development of a new computational method for finding the optimal control which has some unique features. The method appears to be efficient, practical, and flexible, as discussed in Chapter VI. Part of its importance is due to the fact that the method can be extended to a large class of problems, as shown in Chapte r VIII.

Kantorovich's convergence theorem was used to prove that under suitable conditions this computational method can be carried out and converges to the optimal control. The generality and power of the theorem lies partly in the fact that it can be applied to problems in any Banach (complete normed) space. When the solution being sought is a function, a function space is used. This avoids the need for the re - stricted time interval found in recent convergence proofs by McGill and Kenneth[67] and by Kalaba.[47] In addition, Kantorovich's proof allows for oscillation of the iterates and thus covers a larger class of problems than the proof by Kalaba.

A number of numerical results have been found and presented for various plants up to sixth order. This represents a significant addition to the known solutions to fuel optimal control problems. The digital computer program is available and will compute fixed time, fuel optimal control to a given state for any linear, time invariant plant up to tenth order.

Some consideration has been given to the approximations used in mechanizing the calculations on a digital computer, and to the effect of these approximations on the results. Very little about this important but difficult practical aspect appears in the literature on computational methods for finding optimal control.

Finally, the point of view that an iterative approach involves smoothing and/or simplifying the problem in some fashion can be fruitful for future study. In effect this viewpoint has been used by other investigators, but more work needs to be done to provide a logical basis for design of a method and for comparison of the relative effectiveness of different approaches.

An outline of the thesis is given in the next section as a descriptive summary.

F. OUTLINE OF THE THESIS

A method for computing the fixed time, fuel optimal control of a linear time-invariant system to a given final state is examined in some detail. A number of possible extensions, some easy and some not so easy, are suggested in Chapter VIII.

Chapters II and III present the analytical development of the method and some theoretical results concerning it. Chapters IV-VI present the computer program, some of the digital computer results, and a discussion of the experimental aspects of the method.

In Chapter II, linear, time-invariant differential equations are used to describe the system. The result is a sequence of approximate operators which is used as a replacement for the original two-point boundary value problem. This sequence is the one implemented in the digital computer program for numerical studies. It is also used in the theorems of Chapter III.

Chapter III shows some theoretical results, and also outlines some advantages (and disadvantages) of the method.

In Chapter IV the computer program is outlined, and in Chapter V the main computer results are presented. This shows how the method works out in practice.

Chapter VI is devoted to a discussion of the computer results. Some practical strengths and weaknesses of the method are pointed out. A special emphasis is placed on the effect of digital approximations on the results, and on the trade-off between computer time required and the accuracy of the results.

Two problems of a more general nature are outlined in Chapter VII. It can easily be seen how their nonlinear system equations make the result much more difficult to handle.

CHAPTER II

LINEAR PLANT, FIXED TIME, FUEL OPTIMAL CONTROL TO A GIVEN STATE

In this chapter the computational method of the thesis is presented. The analytic results of Chapter III, and the computer program and experiments of Chapters IV-VI are based on the approach developed here. The problem is presented, reduced to a two-point boundary value problem, changed to integral form, replaced by a sequence of approximate integral equations, and made ready for numerical solution by application of Newton's method. Advantages and limitations of this procedure are discussed in Chapter III.

A. PROBLEM 1

Given: a. A system (plant) described by the linear time invariant (vector) differential equation.

$$\dot{\underline{x}}(t) = \underline{A}\,\underline{x}(t) + \underline{b}u(t) \tag{2.1}$$

b. A fixed time interval

$$t \in [0, T] \tag{2.2}$$

c. Initial and terminal boundary conditions on the state vector.

$$\underline{x}(0) = \underline{\xi}$$

$$\underline{x}(T) = \underline{\theta} \tag{2.3}$$

Note: In much of what follows, the terminal state $\underline{\theta}$ is the origin, $\underline{0}$, the equilibrium point of the state equations. In this case, Problem 1 is called a regulator problem.

d. The control variable must satisfy a constraint

$$|u(t)| \leq 1 \quad \text{for all} \quad t \in (0, T] \tag{2.4}$$

Note: The function space of allowable controls $U_{(0, T]}$ from Appendix A is

$$U_{(0,\,T]} = \{u(t):\ |u(t)| \leq 1,\ \text{for all}\ t \in (0, T]\}$$

e. The fuel functional is

$$J(u) = \int_0^T |u(\tau)|\, d\tau \tag{2.5}$$

Then: It is desired to find a control $u^*(t)$ that

a. Satisfies the constraint 2.4.

b. Transfers the system 2.1 from the initial state $\underline{\xi}$ at time $t = 0$ to the terminal state $\underline{\theta}$ at time $t = T$.

c. Minimizes the fuel functional 2.5.

This set of conditions will be called Problem 1.

B. THE TWO-POINT BOUNDARY VALUE PROBLEM*

The relations deduced by applying Pontryagin's Minimum Principle to Problem 1 are summarized below. See Appendix A for a statement of the Minimum Principle.

Definition 2.1: The "deadzone" function $dez[\,\cdot\,]$ is defined as follows:

$$u(t) = dez\,[w(t)] \tag{2.6}$$

means

$$u(t) = 1 \quad \text{when} \quad w(t) > 1$$

$$u(t) = 0 \quad \text{when} \quad |w(t)| < 1$$

$$u(t) = -1 \quad \text{when} \quad w(t) < -1$$

and $u(t)$ is not well defined when $|w(t)| = 1$. The input-output characteristic of the deadzone function is shown in Fig. 2.1.

---

* The two-point boundary value problem will be abreviated to TPBVP.

Let $u^*(t)$, $t \in [0, T]$ be the fuel optimal control, the solution of Problem 1, assuming that one exists. Let $\underline{x}^*(t)$ be the resulting state on the fuel optimal trajectory. Let $\underline{p}^*(t)$, $t \in [0, T]$ be the corresponding costate vector.

Then the Minimum Principle yields the relations:

$$H(\underline{x}^*, u^*, \underline{p}^*, t) = |u^*(t)| + \underline{p}^{*\prime}(t)\,\underline{A}\,\underline{x}^*(t) + \underline{p}^{*\prime}(t)\,\underline{b}\,u^*(t) \tag{2.7}$$

$$\dot{\underline{x}}^*(t) = \frac{\partial H}{\partial \underline{p}^*} = \underline{A}\,\underline{x}^*(t) + \underline{b}\,u^*(t) \tag{2.8}$$

$$\dot{\underline{p}}^*(t) = -\frac{\partial H}{\partial \underline{x}^*} = -\underline{A}'\,\underline{p}^*(t) \tag{2.9}$$

$$\begin{aligned} \underline{x}^*(0) &= \underline{\xi} \\ \underline{x}^*(T) &= \underline{\theta} \end{aligned} \tag{2.10}$$

and the relation

$$H(\underline{x}^*, u^*, \underline{p}^*, t) \leq H(\underline{x}^*, u, \underline{p}^*, t) \text{ for all } u \text{ such that } |u| \leq |$$

yields

$$u^*(t) = -\,\mathrm{dez}\,[\underline{b}'\,\underline{p}^*(t)] \tag{2.11}$$

Comment 2.1: Examination of Eqs. 2.8-2.11 shows that knowledge of $\underline{\pi}^*$, the optimal costate initial condition vector, is sufficient to reduce the TPBVP to an initial value problem (which requires $2n$ straightforward integrations). Determination of $\underline{\pi}^*$ will be considered equivalent to solution of the TPBVP.

C. INTEGRAL EQUATION FORM

The TPBVP of Section B is equivalent to a (vector) Fredholm integral equation of the first kind. In view of Comment 2.1 it reduces to a nonlinear operator on the vector $\underline{\pi}^*$. To see this, first write the solution of Eq. 2.9 .

$$\underline{p}^*(t) = e^{-\underline{A}'t}\,\underline{\pi}^* \quad \text{where} \quad \underline{\pi}^* \triangleq \underline{p}^*(0)$$

Define for convenience

$$\underline{q}(t) \triangleq \underline{e}^{-\underline{A}t}\underline{b} \tag{2.12}$$

then the optimal control 2.11 becomes

$$u^*(t) = -\,dez\left[\underline{b}'\,\underline{e}^{-\underline{A}'t}\underline{\pi}^*\right] = -dez\left[\underline{q}'(t)\underline{\pi}^*\right] \tag{2.13}$$

The solution for the state Eq. 2.8 is

$$\underline{x}^*(t) = \underline{e}^{\underline{A}t}\left[\underline{\xi} + \int_0^t \underline{q}(\tau)u^*(\tau)d\tau\right]$$

If the terminal boundary condition 2.3 is applied, then

$$\underline{e}^{-\underline{A}T}\,\underline{\theta} = \underline{\xi} - \int_0^T \underline{q}(\tau)\,dez\,\left[\underline{q}'(\tau)\underline{\pi}^*\right]\,d\tau$$

For later use with Newton's method, the operator $T(\underline{\pi})$ is defined.

$$T(\underline{\pi}) \equiv \underline{\xi} - \underline{e}^{-\underline{A}T}\,\underline{\theta} - \int_0^T \underline{q}(\tau)\,dez\,\left[\underline{q}'(\tau)\underline{\pi}\right]\,d\tau \tag{2.14}$$

The operator $T(\underline{\pi})$ maps one n dimensional vector into another.

$$T(\underline{\pi}):R_n \rightarrow R_n$$

Problem I is now reduced to finding $\underline{\pi}^*$, the solution vector of the operator equation

$$T(\underline{\pi}^*) = \underline{0} \tag{2.15}$$

For simplicity, $\underline{\pi}^*$ will be referred to as the solution of the operator $T(\underline{\pi})$. Also, in most of what follows the final state is the orgin, so $T(\underline{\pi})$ becomes

$$T(\underline{\pi}) = \underline{\xi} - \int_0^T \underline{q}(\tau)\,dez\,\left[\underline{q}'(\tau)\underline{\pi}\right]\,d\tau \tag{2.16}$$

D. SEQUENCE OF APPROXIMATE EQUATIONS

A sequence of approximate operators $\{T_k(\underline{\pi})\}$ is now introduced to replace the operator $T(\underline{\pi})$. The idea is to start with a very simple operator and work up by steps toward the exact operator $T(\underline{\pi})$. By doing this properly, Newton's method can be guaranteed to converge at each step, so that a workable computational approach results. Two approximations will be introduced; one is a linear term to get the computations started successfully, and the other is a sequence of smooth functions $u_k(\cdot)$ with a parameter η_k, $k=0, 1, 2 \ldots k_1$. As $\eta_k \rightarrow \infty$, $u_k(\cdot) \rightarrow u^*(\cdot)$. So the idea is to start with a linear approximation ($\eta_0=0$), then to drive the linear part to zero and increase η_k so that the approximate control $u_k(\cdot)$ converges to the optimal control $u^*(\cdot)$.

When the optimal control $u^*(\underline{q}'(\tau)\underline{\pi}^*)$ is replaced by u_k the form of the optimal control argument $\underline{q}'(\tau)\,\underline{\pi}$ will be retained.

The simplest useful control one could start with is a linear one.

Change 1. First apply a linear control

$$u_0(\cdot) = \alpha_0(\cdot)$$

Using the control argument $\underline{q}'(t)\underline{\pi}$ yields

$$u_0(\underline{q}'(t)\underline{\pi}) = \alpha_0\underline{q}'(t)\underline{\pi}$$

Inserting this control into the differential Eq. 2.1 and applying the given boundary conditions leads to the zeroeth approximate operator.

$$T_0(\underline{\pi}) = \underline{\xi} - \int_0^T \underline{q}(\tau)\alpha_0\underline{q}'(\tau)\underline{\pi}d\tau$$

Let $\underline{W}(T)$ be the controllability matrix

$$\underline{W}(T) = \int_0^T \underline{q}(\tau)\underline{q}'(\tau)d\tau \qquad (2.17)$$

Then

$$T_0(\underline{\pi}) = \underline{\xi} - \alpha_0\underline{W}(T)\underline{\pi}$$

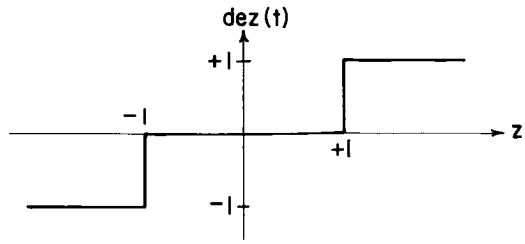


Fig. 2.1 The Deadzone Function dez

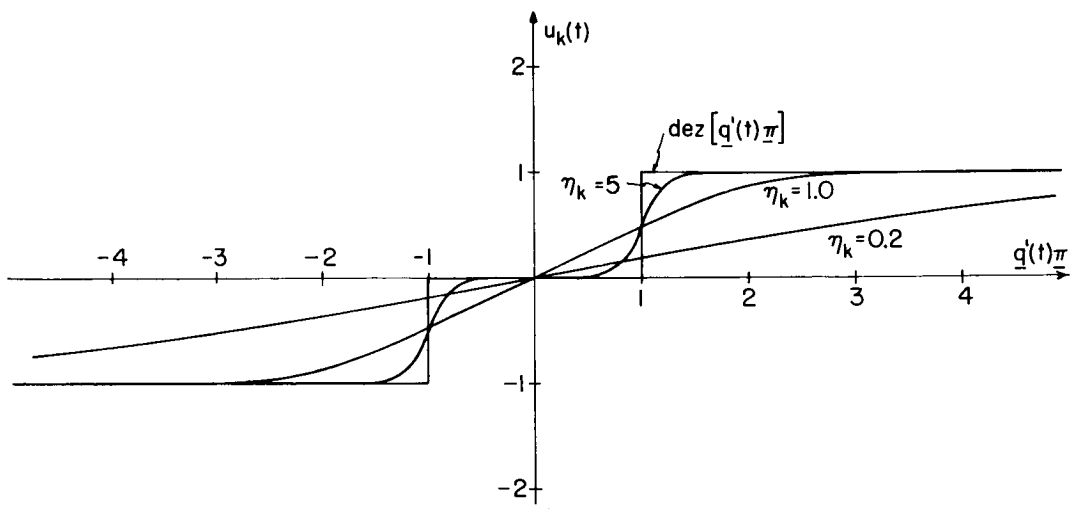


Fig. 2.2 The Approximate Control Function u_k

In order to approximate the optimal control function, an exponential (actually hyperbolic tangent) form $u_k(\cdot)$ is introduced with a scalar approximation factor η_k. The deadzone function can be approximated as closely as desired (where it is defined) by an analytic function, since the points of discontinuity are excluded (see Dieudonne[25] for a simple treatment of this).

Change 2. Introduce an approximate control function $u_k(\cdot)$.

$$u_k(\cdot) = \frac{1}{2} \{\tanh [\eta_k(\cdot + 1)] + \tanh [\eta_k(\cdot - 1)] \}$$

Using the control argument $\underline{q}'(t)\underline{\pi}$ yields

$$u_k(\underline{q}'(t)\underline{\pi}) = \frac{1}{2} \{\tanh [\eta_k(\underline{q}'(t)\underline{\pi} + 1)] + \tanh [\eta_k(\underline{q}'(t)\underline{\pi} - 1)]\} \qquad (2.18)$$

A plot of $u_k(t)$ as a function of $\underline{q}'(t)\underline{\pi}$ is shown in Fig. 2.2 for some typical values of η_k. As η_k increases, $u_k(t)$ approaches the deadzone function $u^*(t)$.

The general approximate operator uses both of the above changes.

$$T_k(\underline{\pi}) = \underline{\xi} - a_k \underline{W}(T)\underline{\pi} - \int_0^T \underline{q}(\tau) u_k(\underline{q}'(\tau)\underline{\pi}) d\tau \qquad (2.19)$$

Let the sequence of approximate operators have k_1 members or steps. Then in order to approximate $u^*(\cdot)$ the slope a_k should be driven to zero and η_k should be increased at each step until some suitably large limit η_{k1} is reached.

$$0 < \eta_1 < \eta_2 \ldots < \eta_{k_1} < \infty$$

$$a_0 > \ldots > a_{k_2} > a_{k_2+1} = \ldots = a_{k_1} = 0 \qquad (2.20)$$

where $k_2 < k_1$

In Fig. 2.3 the sum of the two changes is shown for a typical sequence of approximate operators. This sequence has three members ($k_1=3$), and one of them contains a linear term ($k_2=2$).

Define the vector $\underline{\pi}_k$ to be the solution vector of the $k^{\underline{th}}$ operator equation. That is,

$$T_k(\underline{\pi}_k) = \underline{0}$$

Definition 2.2: Applied sequentially means the solution vector $\underline{\pi}_{k-1}$ of the previous operator $T_{k-1}(\underline{\pi})$ is used as a starting vector for Newton's method on the present operator $T_k(\underline{\pi})$.

Properties of the sequence are discussed in Chapter III, Section D, but as an introduction the main points are listed here.

1. A sequence can be found such that Newton's method converges when applied to each member sequentially.
2. Under suitable restrictions this sequence of operators converges (in the L_2 norm) to the exact operator $T(\underline{\pi})$.
3. The solutions to the approximate operators lead to suboptimal controls which use only a little more fuel than the optimal control, yet do not require the instantaneous switching of the optimal control.

It only remains to show what size steps to make in the parameters η and α. The aim is to make these steps large, yet still guarantee that Newton's method will converge.

Definition 2.3: Assume the solution vector $\underline{\pi}_k$ of the operator $T_k(\underline{\pi})$ has been found. Now make changes $\Delta\eta$ and $\Delta\alpha$ in the parameters η and α to form a new operator T_{k+1}. Apply Newton's method to T_{k+1} sequentially (by Definition 2). The set of all changes $\Delta\eta$ and $\Delta\alpha$ such that Newton's method converges (when applied to T_{k+1}) is called the region of convergence, about η_k and α_k in the parameter space. There is a corresponding region of convergence in the space $\underline{\pi}$ of solution vectors.

A short experimental investigation was made of the region of convergence for a typical problem of the class to be studied, and Fig. 2.4 gives some idea of the results.

In Fig. 2.4 a typical sequence of the solution vectors $\{\underline{\pi}_k\}$ is plotted with the region of convergence indicated for each vector $\underline{\pi}_k$. The starting vector $\underline{\pi}_0$ can always be found. Then the region of convergence gradually decreases as the sequence approaches the

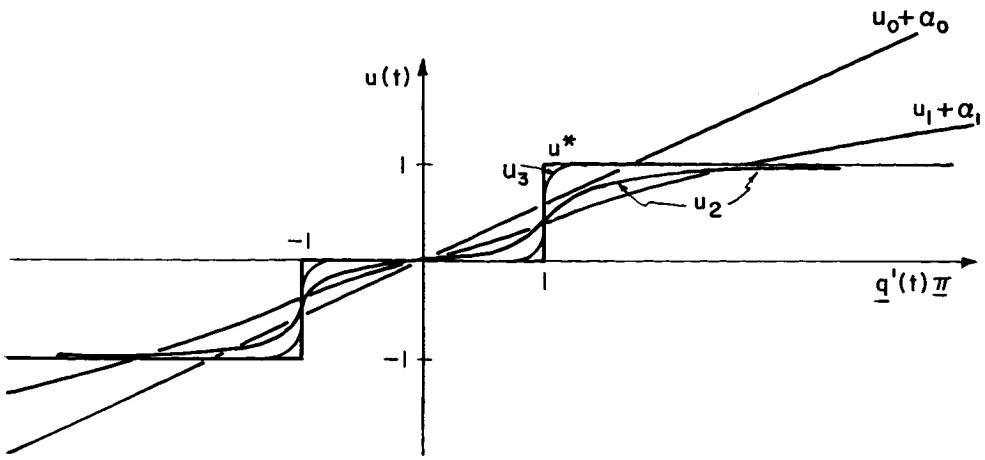


Fig. 2.3 Typical Sequence of Approximate Controls (The Sum of u_k and the Linear Term)

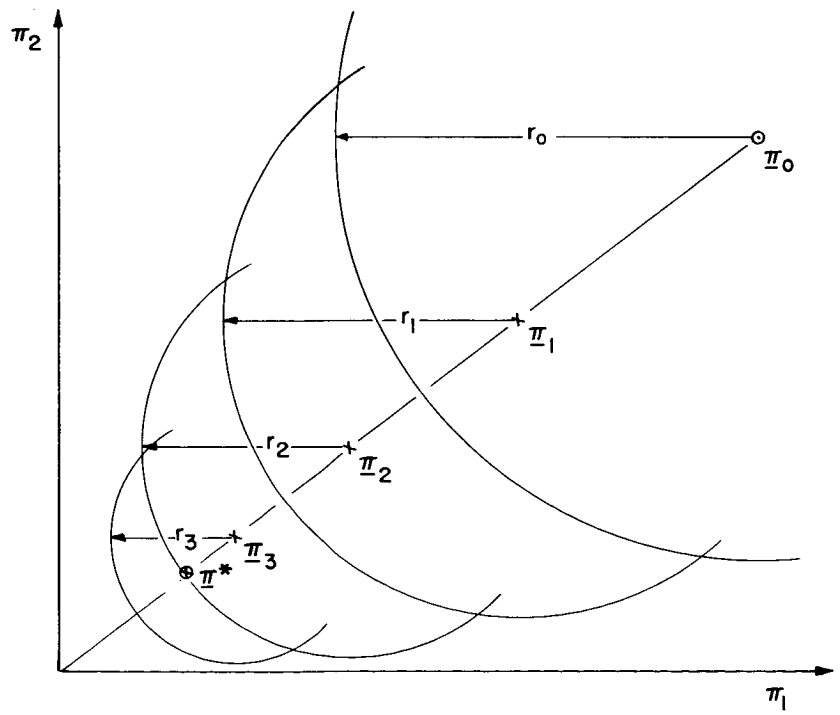


Fig. 2.4 Regions of Convergence for the Sequence $\{\underline{\pi}_k\}$

exact operator. However, the region of convergence soon includes the exact solution vector $\underline{\pi}^*$ as shown in Fig. 2.4.

This experimental investigation also indicates it is best not to get too close to the edge of the region of convergence. Newton's method tends to oscillate and make only small progress per step toward the solution when the iterate is near this edge (in the ℓ_2 norm). Thus if total computer time for the calculations is to be kept low, it is good to stay well inside the region of convergence, even if the sequence then has more members. Also, this gives a factor of safety in case the size of the region of convergence has been overestimated.

An attempt was made in Section F to find an analytic estimate for the maximum size of these steps, using the convergence theorem of Kantorovich and Akilov[49], page 708. This was not satisfactory because:

1. The effect of η and α on the required norms was too complicated, and
2. The sufficient conditions are too conservative, especially as η becomes large.

Due to the exponential nature of the approximate control function, it seems logical to increase η at an exponential rate. Further, the ratio $\frac{\eta_{k+1}}{\eta_k}$ used is affected by the number of dimensions (n) of the state space. In the experimental investigation of the region of convergence, an effort was made to see if an equation of the form

$$\left(\frac{\Delta\eta}{\eta}\right)^{c_1} n^{c_2} = c_3$$

c_1, c_2, c_3 constants

might adequately predict a good step size in η. Good agreement was indicated ($\Delta\eta$ large, but not too near the limit of the region of convergence) for

$$c_1 = 4$$

$$c_2 = 3$$

$$c_3 = 60$$

The resulting formula for changing η is

$$\Delta \eta_k = \left(\frac{60}{n^3}\right)^{1/4} \eta_k \tag{2.21}$$

and of course

$$\eta_{k+1} = \eta_k + \Delta \eta_k \tag{2.22}$$

Now $\Delta \eta$ is fixed but Δa still must be chosen. Remember from the sequence 2.20 that it is desired to reduce a quickly in order to get rid of the extra linear term introduced to start the sequence. First order estimates are made of the effect of the control u_{k+1} on the state. Then a proportion rule is used to choose Δa.

$$\frac{\Delta a}{a} = \frac{\text{effect of control } u_{k+1}}{\text{effect of the linear term}} \tag{2.23}$$

In the first member of the sequence the linear term causes a change in the state vector, given by $\underline{\xi} - \underline{e}^{-\underline{A}^T}\underline{\theta}$. This expression is used as an upper bound estimate on the effect of the linear term.

To estimate the effect of the approximate control u_{k+1} on the state vector the previous solution $\underline{\pi}_k$ is used, and a vector $\underline{g}_k$ is defined.

$$\underline{g}_k = \int_0^T \underline{q}(\tau)\, u_{k+1}[\underline{q}'(\tau)\underline{\pi}_k]\, d\tau \tag{2.24}$$

These vector estimates are normed and substituted into the proportion (2.23). A $(1 + \eta)$ factor is included to accelerate the changes in a if η is large.

$$\Delta a_k = -a_0 \frac{\|\underline{g}_k\|_1}{\|\underline{\xi} - \underline{e}^{-\underline{A}^T}\underline{\theta}\|_1} (1 + \eta_k) \tag{2.25}$$

and

$$a_{k+1} = \max\{0,\ a_k + \Delta a_k\} \tag{2.26}$$

Formulas 2.21 and 2.25 yield changes $\Delta\eta$ and Δa which were found to work in the examples studied. In most cases, studied during the course of this research and presented in Chapter V, Newton's method converged when applied sequentially to the new operator T_{k+1}.

E. APPLYING NEWTON'S METHOD

Newton's method is to be applied to a typical operator $T_k(\underline{\pi})$. Newton's method is covered in Appendix B, but a short introduction is given below. Given the operator Eq. 2.19, to find the solution vector $\underline{\pi}_k$ such that

$$T_k(\underline{\pi}_k) = 0,$$

one linearizes about the current guess $\underline{\pi}^i$

$$T_k(\underline{\pi}_k) \approx T_k(\underline{\pi}^i) + (\underline{\pi}_k - \underline{\pi}^i)\, T_k^{(1)}(\underline{\pi}^i)$$

Then the next iterate is found by solving this linear equation for $\underline{\pi}_k$.

$$\underline{\pi}^{i+1} = \underline{\pi}^i - \left[T_k^{(1)}(\underline{\pi}^i)\right]^{-1} T_k(\underline{\pi}^i) \qquad (2.27)$$

Equation 2.27 is the recursive relation of Newton's method. Since T_k has a vector valued range space, its first derivative is the Jacobian matrix.

$$T_k^{(1)}(\underline{\pi}) = -a_k \underline{W}(T) - \int_0^T \underline{g}(\tau)\, \underline{g}'(\tau)\, u_k^{(1)} \left[\underline{g}'(\tau)\underline{\pi}\right] d\tau$$

Then Eq. 2.27 can be written out entirely in matrix notation.

$$\underline{\pi}^{i+1} = \underline{\pi}^i + \left[a_k \underline{W}(T) + \int_0^T \underline{g}(\tau)\, \underline{g}'(\tau)\, u_k^{(1)} \left[\underline{g}'(\tau)\underline{\pi}^i\right] d\tau\right]^{-1} \left[\underline{\xi} - a_k \underline{W}(T)\underline{\pi}^i\right.$$

$$\left. - \int_0^T \underline{g}(\tau) u_k \left[\underline{g}'(\tau)\underline{\pi}^i\right] d\tau\right] \qquad (2.28)$$

The approximate control function u_k is, from Eq. 2.18,

$$u_k[\underline{q}'(\tau)\underline{\pi}] = \frac{1}{2}\{\tanh[\eta_k(\underline{q}'(\tau)\underline{\pi} + 1)] + \tanh[\eta_k(\underline{q}'(\tau)\underline{\pi} - 1)]\} \tag{2.29}$$

and its first derivative is,

$$u_k^{(1)}[\underline{q}'(\tau)\underline{\pi}] = \frac{1}{2}\eta_k\{2\text{-}\tanh^2[\eta_k(\underline{q}'(\tau)\underline{\pi} + 1)] - \tanh^2[\eta_k(\underline{q}'(\tau)\underline{\pi} - 1)]\} \tag{2.30}$$

This recursive relation 2.28 with Eqs. 2.29 and 2.30 will be called the inner loop. Starting with an inital guess $\underline{\pi}^0$, Eq. 2.28 is applied repeatedly. If at some step i, $\underline{\pi}^i \approx \underline{\pi}^{i-1}$, the inner loop is said to have converged, and the vector $\underline{\pi}^i$ is defined to be the solution vector $\underline{\pi}_k$ of the operator T_k.

F. CONDITIONS FOR CONVERGENCE

A sufficient condition for the convergence of Newton's method has been given by Kantorovich[49], page 708. This condition is shown in Appendix B as Theorem B.1. The purpose of this section is to apply the sufficient condition to the approximate operator $T_k(\underline{\pi})$ of 2.19 and to write out the required expressions.

The first two derivatives of $T_k(\underline{\pi})$ are given in Chapter III as Eqs. 3.4 and 3.5. Here the task is to evaluate or bound certain norms.

Let
$$\Gamma_{k+1} \triangleq [T_{k+1}^{(1)}(\underline{\pi}_k)]^{-1} \tag{2.31}$$

Then

$$A_{norm} = \|\Gamma_{k+1} T_{k+1}(\underline{\pi}_k)\|$$

$$B_{norm} = \|\Gamma_{k+1} T_{k+1}^{(2)}(\underline{\pi})\|$$

are the required norms. B_{norm} is to be evaluated over all possible vectors $\underline{\pi}$ belonging to the n dimensional vector space.

Using the definition of the operator T_k and its two derivatives leads to the expanded formulas

$$\Gamma_{k+1} = -\left[\alpha_{k+1}\underline{W}(T) + \int_0^T \underline{g}(\tau)\underline{g}'(\tau)u_{k+1}^{(1)}\left[\underline{g}'(\tau)\underline{\pi}_k\right] d\tau\right]^{-1} \quad (2.32)$$

$$A_{norm} = \left\| \Gamma_{k+1} \cdot \left[\underline{\xi} - \alpha_{k+1}\underline{W}(T)\underline{\pi}_k - \int_0^T \underline{g}(\tau)u_{k+1}\left[\underline{g}'(\tau)\underline{\pi}_k\right] d\tau\right]\right\| \quad (2.33)$$

$$B_{norm} = \left\| \Gamma_{k+1} \cdot \int_0^T \underline{g}(\tau)\underline{g}'(\tau)\underline{y}\,\underline{g}'(\tau)\underline{z}\; u_{k+1}^{(2)}\left[\underline{g}'(\tau)\underline{\pi}\right] d\tau\right\| \quad (2.34)$$

There remains the problem of searching over the space P of costate initial condition vectors for the one which yields the largest value of the norm. Actually it is only necessary to search in a sphere around $\underline{\pi}_k$, but the radius of this sphere is not known beforehand. To simplify matters, an upper bound is established by using the maximum value which $u^{(2)}(\cdot)$ can assume.

$$B_{norm} \leq \left\| \Gamma_k \cdot \int_0^T q(\tau)q'(\tau)yq'(\tau)z\, d\tau \cdot u_{max}^{(2)} \right\| \quad (2.35)$$

The expression inside A_{norm} is a vector, so any vector norm can be used. The ℓ_∞ or maximum norm is chosen for simplicity and also because it yields the smallest result of any of the ℓ_p (Lebesque) norms for a vector.

Thus if the argument of 2.32 is a vector s_i,

$$A_{norm} = \left\| \underline{s}_i \right\|$$

one can use

$$A_{norm} = \max_i \left| \underline{s}_i \right| \quad (2.36)$$

For B_{norm}, a third-order tensor must be handled. In this case, the argument of the norm is of the form S_{ijk}.

$$B_{norm} = \left\| S_{ijk} \right\|$$

In Appendix B it is shown that B_{norm} can be bounded by the expression

$$B_{norm} \leq \max_i \sum_j \sum_k |S_{ijk}| \qquad (2.37)$$

Kantorovich's theorem then guarantees convergence if

$$h \triangleq A_{norm} \cdot B_{norm} < 1/2 \qquad (2.38)$$

One of the research aims was to see how closely Eq. 2.38 would predict the actual extreme conditions for convergence. Also it was hoped that this information could be used in designing the sequence of approximate operators. Toward this end, another norm similar to B_{norm} was defined which was intended to provide a closer estimate of the actual limit of convergence.

$$C_{norm} = \| \Gamma_{k+1} T^{(2)}_{k+1} (\underline{\pi}_k) \|$$

and in expanded form,

$$C_{norm} = \| \Gamma_{k+1} \int_0^T \underline{g}(\tau)\underline{g}'(\tau)\underline{y}\, \underline{g}'(\tau)\underline{z}\, u^{(2)}_{k+1} [\underline{g}'(\tau)\underline{\pi}_k]\, d\tau \|$$

$$h_1 = A_{norm} \cdot C_{norm} \qquad (2.39)$$

The idea is to see whether or not the expression $A_{norm} \cdot C_{norm}$ more accurately predicts the limit of convergence for Newton's method

In Chapter IV, Section I a way of approximating these expressions on a digital computer is shown. In Chapter V some numerical experiments were performed in finding the parameters h and h_1. Both the guaranteed convergence parameter h and the estimated convergence parameter h_1 turned out to have values far in excess of 1/2, although h_1 was usually one or two orders of magnitude smaller than h. The conclusion is that the convergence theorem is very useful in theoretical studies, but much too conservative to give practical estimates of the region of convergence.

G. SUMMARY

The original minimization problem was converted to a two-point boundary value problem. This was put into integral form and reduced to a nonlinear vector relation (or operator); i.e., the problem is considered solved once the initial costate vector $\underline{\pi}^*$ is found. The vector relation was replaced by a sequence of approximate vector operators. A method was designed for choosing the sequence so that Newton's method could be applied to it, and so that it approached as closely as desired to the true solution. These properties are verified experimentally in Chapter V and analytically in Chapter III.

The procedure consists of applying Newton's method sequentially to the sequence of approximate operators to determine their solution vectors $\{\underline{\pi}_k\}$. These vectors lead to a sequence of approximate controls which converge (as closely as desired) to the optimal control, and which in Chapters III and V are shown to have unique properties of their own as suboptimal controls.

CHAPTER III

ADVANTAGES, DISADVANTAGES, AND LIMITATIONS

In this chapter some of the properties of the procedure outlined in Chapter II are discussed. The discussion is divided into sections parallel to those of Chapter II, to give insight into each step in the development of the method. It is shown that under suitable assumptions the suggested method converges to the optimal control.

A. PROBLEM 1

The fixed-time, fixed-terminal state problem with linear, time-invariant plant, limited control effort, and total fuel cost criterion represents one specific class of problems out of many possible ones. It was desired to pick a class of problems to examine in some depth, and this is a particularly apt one. Some possible extensions are described in Chapters VII and VIII.

B. PONTRYAGIN'S MINIMUM PRINCIPLE (Two-Point Boundary Value Problem)*

The theorem provides a set of necessary, but not sufficient conditions for a minimum to exist. Thus if there is no solution to the TPBVP, then no optimal control exists. This would occur for instance if the terminal state $\underline{\theta}$ were not reachable at time T from the initial state $\underline{\xi}$. However, the existence of one, or even several, solutions to the TPBVP does not in general guarantee the existence of an optimal control.

A solution of the TPBVP corresponds to a stationary point of the original problem; a maximum, a minimum, or an inflection point. The problem faced by the user then is this: Given a solution to the TPBVP, how can one be sure it leads to the absolute minimum of the cost functional? If one finds more than one solution, the cost functionals can be computed and compared directly to decide between

---

* The abbreviation TPBVP will be used for the Two Point Boundary Value Problem.

them. In general, though, only an appeal to the "reasonableness of the solution" can be made. Does the solution seem reasonable physically? Does it yield a lower cost than other control functions that have been tried or used?

Analytic Sufficiency Conditions for Optimality

For Problem 1 there are some results available, under the following additional conditions:

Assumption 1: The System 2.1 is controllable.

Assumption 2: The state $\underline{\theta}$ is reachable at time T from the state $\underline{\xi}$ (see Appendix A for a definition of reachability). Essentially this means that there exists at least one allowable control transferring the system from the state $\underline{\xi}$ to the state $\underline{\theta}$ in time T.

Assumption 3: The system matrix of 2.1 is nonsingular, i.e.,

$$\det \underline{A} \neq 0 \tag{3.1}$$

Then: a. Assumption 2 is sufficient to guarantee the existence of a fuel-optimal control for Problem 1 (see Neustadt[72]).

b. Assumptions 1 and 3 are sufficient to guarantee that the fuel-optimal control will be normal. That is, the argument of the optimal control relation 2.11 cannot remain constant over any finite time interval. Thus, given any solution of the costate equations 2.9 the corresponding control from relation 2.11 is specified almost everywhere (see for example Athans and Falb,[4] pp. 443-445).

As a consequence of normality, two theorems have been proven by Athans and Falb,[4] pp. 445-450.

1. The optimal control for Problem 1 (if one exists) is unique.

2. Moreover, the extremal control (if one exists) is unique.

Note that Assumptions 1 and 3 are satisfied for most of the examples investigated in Chapter V. Then if a solution to the TPBVP

exists, it is the optimum solution to Problem 1; i.e., if the procedure converges, it converges to the optimum initial costate vector $\underline{\pi}^*$.

The remaining parameter is how <u>difficult</u> the problem is, ranging from easy through increasingly difficult to impossible. A problem is impossible if Assumption 2 is violated. The implications of this degree of difficulty of the problem are discussed in Chapter VI.

C. INTEGRAL EQUATION FORM

In the integral form the state equations, costate equations, and boundary conditions are combined in a single set of equations. This is convenient for redefining the problem as an operator equation. Also the integral form shows explicitly how the problem reduces to a search for the costate initial condition vector $\underline{\pi}^*$. In fact, in Chapter II the integral form really results from the attempt to solve Problem 1 in a general way.

The integral form does not introduce any new difficulties. Clearly if the final state is not reachable (cf. Assumption 2 above) the resulting operator will not possess a zero solution (for either Problem 1 of Chapter II or Problem 2 of Chapter VII).

Recognizing the costate initial condition vector $\underline{\pi}$ as the unknown variable, however, <u>does</u> add a new complication; namely that there may exist more than one vector $\underline{\pi}$ leading to the same control $\underline{u}(t)$ for $t \epsilon [0, T]$. In the case of an optimal or an extremal control $\underline{u}^*(t)$ in <u>Problem 1</u>, a simple condition can be stated for the uniqueness of $\underline{\pi}^*$.

1. Let $\underline{\pi}^*$ be a costate initial condition vector, which leads, using the deadzone operator, to a given optimal control $u^*(t)$.

2. Let the condition

$$|\underline{q}'(t)\, \underline{\pi}^*| = 1 \qquad (3.2)$$

occur at times $t = t_1, t_2 \ldots t_m$ in the open interval $(0, T)$.

The switch times are assumed distinct. This will be true unless the problem is singular.

Lemma 3.1: A necessary and sufficient condition for the uniqueness of $\underline{\pi}^*$, given $\underline{u}^*(t)$, is that the set of vectors $\underline{q}(t_i)$, $i = 1, 2, \ldots m$ span the space R_n. That is, the matrix $\underline{Q}$ must have maximal rank (rank n), where

$$\underline{Q} = \begin{bmatrix} \vdots & \vdots & & \vdots \\ \underline{q}(t_1), & \underline{q}(t_2), & \ldots\ldots, & \underline{q}(t_m) \\ \vdots & \vdots & & \vdots \end{bmatrix} \tag{3.3}$$

Proof: Since the m vectors $\underline{q}(t_i)$ span the space R_n, a valid basis for the space can be found among them. Specifying the projections of a vector $\underline{\pi}$ in the direction of the basis vectors specifies a unique vector $\underline{\pi}^*$. From Eq. 3.3, each of these lengths or projections is ± 1.

Conversely, if the m vectors $\underline{q}(t_i)$ do not span the space R_n, a basis can be formed by adding one or more properly chosen additional vectors $\underline{q}_j$ in the directions not covered by the vectors $\underline{q}(t_i)$. Now there is a basis, but one or more of the lengths in it is arbitrary. For each different length in the $\underline{q}_j$ direction(s) a different vector $\underline{\pi}$ is specified.

Comment 3.1: In the second case, if the origin of R_n is translated to the point specified by taking zero as the length(s) in the $\underline{q}_j$ direction(s), then the vector $\underline{\pi}$ is constrained to lie in a subspace of the translated R_n. The dimension of this subspace would be determined by the rank of the matrix $\underline{Q}$, i.e., n-rank of $\underline{Q}$ = dimension of the region of $\underline{\pi}$.

D. SEQUENCE OF APPROXIMATE EQUATIONS

As stated in Chapter II, the sequence of approximate equations is chosen to:

1. provide a set of useful suboptimal controls,
2. yield convergence when Newton's method is applied to them sequentially,

3. converge to the original equations.

Here it is shown in what sense the sequence possesses these properties. Problem 1 is used in the lemmas and theorems, some of which might also hold for a more general problem.

1. Suboptimal Controls

The easiest property to show is the first one above. In Problem 1 for instance, once the linear term has been eliminated ($a \to 0$) each member of the sequence has a solution which leads to a feasible control. A feasible control is one which transfers the state from the initial condition $\underline{\xi}$ to the final condition $\underline{\theta}$ in T seconds and does not violate magnitude constraints on u(t).

Further, note in Fig. 2.2 that the approximate function chosen to replace the deadzone function is a good approximation except when the argument is near zero. In the optimal control equations the costate trajectory, also called the influence function, measures the effectiveness of the control in reducing the penalty function J. To be more specific, at any time $t_1 \epsilon (0, T)$ the argument of the control function

$$\underline{b}'\underline{e}^{-\underline{A}'t_1}\underline{\pi}^* = \underline{q}'(t_1)\underline{\pi}^*$$

gives the total per unit effectiveness of $u^*(t_1)$ in reducing the cost functional J. Using the approximate control function instead of the deadzone function requires some change in the costate initial condition (that is $\underline{\pi}_k \neq \underline{\pi}^*$), but to first order the influence functions retain this property.

Thus an approximation which is bad only where the argument is small should not increase the cost by much - i.e., should be an efficient suboptimal control. How efficient this turns out to be will be seen in the computer examples.

Notice also that in a rough way this approximates the effect of simulating the optimal control by a physical control having dispersion (for a definition of dispersion see Paynter[101]). This is somewhat like using a relay whose output is connected to a transmission

line as the actuating element. Relays have rise times, and transmission lines tend to smooth the function out. So the suboptimal controls chosen have at least some relation to the problem of designing a near optimal control.

In summary, the suboptimal controls are feasible, efficient, and somewhat practical.

2. Sequential Convergence

The basic convergence theorem, Theorem B.1 of Appendix B will be applied to Problem 1. Under enough suitable restrictions a similar result would hold in the more general case of Problem 2.

Operator 2.19 is the expression to be examined.

$$T_k(\underline{\pi}) = \underline{\xi} - a_k \underline{W}(T)\,\underline{\pi} - \int_0^T \underline{q}(\tau) u_k[\underline{q}'(\tau)\underline{\pi}]\,d\tau \tag{2.19}$$

As mentioned in Chapter II, this is a vector expression with a vector argument, $T_k : R_n \to R_n$. To be more exact, the range of the operator is a closed subset of R_n consisting of the set of reachable states at time T. The derivative as defined in Appendix B is a linear operation $T_k^{(1)}\underline{y} : R_n \to R_n$, and it employs a dummy vector variable $\underline{y}$ in order to retain this vector characteristic. Applying the definition of derivative yields the linear operator 3.4.

$$T_k^{(1)}(\underline{\pi})\,\underline{y} = -a_k \underline{W}(T)\,\underline{y} - \int_0^T \underline{q}(\tau)\underline{q}'(\tau)\underline{y}\, u_k^{(1)}[\underline{q}'(\tau)\underline{\pi}]\,d\tau \tag{3.4}$$

In the same way, the second derivative is a bilinear operator with two dummy variables $\underline{y}$ and $\underline{z}$

$$T_k^{(2)}(\underline{\pi})\underline{y}, \underline{z} = -\int_0^T \underline{q}(\tau)\underline{q}'(\tau)\underline{y}\,\underline{q}'(\tau)\underline{z}\, u_k^{(2)}[\underline{q}'(\tau)\,\underline{\pi}]\,d\tau \tag{3.5}$$

where $u^{(1)}[\cdot]$ is the ordinary derivative of a function u with respect to its argument $\underline{q}'(\tau)\underline{\pi}$. In the ordinary (as opposed to

functional) way of taking the vector derivative, the first derivative is a matrix and the second derivative is a third order tensor. To show that the result is essentially the same, note that the dummy variable $\underline{y}$ can be removed from the integral of Eq. 3.4. The result is

$$T_k^{(1)}(\underline{\pi}) = -a_k \underline{W}(T) - \int_0^T \underline{q}(\tau)\underline{q}'(\tau) u_k^{(1)}[\underline{q}'(\tau)\underline{\pi}]\, d\tau \tag{3.6}$$

$$= -\int_0^T \underline{q}(\tau)\underline{q}'(\tau)\{a_k + u_k^{(1)}[\underline{q}'(\tau)\underline{\pi}]\} d\tau$$

Operator 3.6 is the Jacobian matrix, the result of taking the ordinary derivative of operator 2.18.

The approximate control u_k and its first derivative $u_k^{(1)}$ have been written out completely as Eqs. 2.29 and 2.30. The second derivative $u_k^{(2)}$ is shown below.

$$u_k^{(2)}[\underline{q}'(\tau)\underline{\pi}] = -\eta_k^2 \{\tanh[\eta_k(\underline{q}'(\tau)\underline{\pi}+1)] - \tanh^3[\eta_k(\underline{q}'(\tau)\underline{\pi}+1)]$$

$$+ \tanh[\eta_k(\underline{q}'(\tau)\underline{\pi}-1)] - \tanh^3[\eta_k(\underline{q}'(\tau)\underline{\pi}-1)]\} \tag{3.7}$$

Note: In what follows, the absolute value of a vector or matrix is taken to mean the vector consisting of the absolute value of each of the components. Thus

$$|\underline{q}'(t)| \equiv [\,|q_1(t)|\ \ |q_2(t)|\ldots|q_n(t)|\,]$$

Lemma 3.2:

The operator 2.18 and its first two derivatives have finite norms, for all finite values of η_k.

For the operator this means $\|T_k(\underline{\pi})\|$ is finite if $\|\underline{\pi}\|$ is finite. For the derivatives it means $\|T_k^{(2)}(\underline{\pi})\underline{y}, \underline{z}\|$ is finite if $\|\underline{y}\|$ and $\|\underline{z}\|$ are finite, and $\|T_k^{(1)}(\underline{\pi})\underline{y}\|$ is finite if $\|\underline{y}\|$ is finite.

Proof: 1. The operator $T_k(\underline{\pi})$

For any value of $\underline{q}'(\tau)\underline{\pi}$ and any value of η, u_k satisfies $|u_k| \leq 1$. Then there follows,

$$\| T_k(\underline{\pi}) \| \leq \| \underline{\xi} - a_k \underline{W}(T)\underline{\pi} \| + \| \int_0^T |\underline{q}(\tau)\, d\tau \|$$

which yields

$$\| T_k(\underline{\pi}) \| \leq |a_k| \cdot \| \underline{W}(T) \| \cdot \| \underline{\pi} \| + \| \underline{\xi} \| + \| \int_0^T |\underline{q}(\tau)\, d\tau \| \qquad (3.8)$$

The fundamental matrix has finite entries, so the vector integral and the controllability matrix both have finite elements. Then assuming the added linear term a_k, the initial state vector, and the number of dimensions n of the state vector are finite, it follows that the operator 2.19 has finite norm whenever the initial costate vector $\underline{\pi}$ does.

Note further that as soon as the extra linear term has been eliminated by driving a_k to zero, the operator has finite norm independent of $\underline{\pi}$.

2. The first and second derivatives

In either case, for any finite value of η_k the derivatives $u_k^{(1)}$ and $u_k^{(2)}$ are continuous, bounded functions with well defined maximum values. Let

$$|u_k^{(1)}| \leq c_4 \quad \text{for all values of } \underline{q}'(\tau)\underline{\pi}$$

$$|u_k^{(2)}| \leq c_5 \quad \text{for all values of } \underline{q}'(\tau)\,\underline{\pi}$$

Since the dummy variables are to have finite norm, they may as well have a norm of 1. Thus we set $\|\underline{y}\| = \|\underline{z}\| = 1$. It follows that the scalar function $\underline{q}'(t)\underline{y}$ is finite, since the elements of $\underline{q}(t)$ are finite. Using the supremum norm over time along with any of the l_p norms yields a bound on the norm of $\underline{q}'(t)\underline{y}$ when $\|\underline{y}\| = 1$. One such bound is

$$\|\underline{q}'(t)\underline{y}\| \leq \sup_{t \in [0, T]} \sum_{i=1}^{n} |q_i(t)|$$

For convenience two constants are defined which depend only on the plant (the state equations) and the time interval $[0, T]$. Let

$$c_6 = \sup_{t\epsilon[0,T]} \sum_{i=1}^{n} |q_i(t)|$$

and

$$c_7 = \left\| \int_0^T |\underline{q}(\tau)| \, d\tau \right\|$$

The rest follows almost by inspection. For the second derivative operator the norm can be bounded.

$$\|T_k^{(2)}(\underline{\pi})\| = \max_{\substack{\|\underline{y}\|=1, \\ \|\underline{z}\|=1}} \|T_k^{(2)}(\underline{\pi})\underline{y}, \underline{z}\|$$

$$= \max_{\substack{\|\underline{y}\|=1 \\ \|\underline{z}\|=1}} \int_0^T \|\underline{q}(\tau)\underline{q}'(\tau)\underline{y} \; \underline{q}'(\tau) \, \underline{z} \, u_k^{(2)}[\underline{q}'(\tau)\underline{\pi}] \, d\tau \|$$

and

$$\|T_k^{(2)}(\underline{\pi})\| \leq \left\| \int_0^T |\underline{q}(\tau)| \, d\tau \right\| \cdot \|\underline{q}'(t)\underline{y}\|^2 \cdot \sup |u_k^{(2)}(\cdot)|$$

$$\leq c_7 c_6^2 c_5 \tag{3.9}$$

The right hand side of Eq. 3.9 is obviously finite.

Applying a similar reduction to the first derivative operator results in,

$$\|T_k^{(1)}(\underline{\pi})\| \leq |a_k| \cdot \|\underline{W}(T)\| + \left\| \int_0^T |\underline{q}(\tau)| \, d\tau \right\| \cdot \|\underline{q}'(t)\underline{y}\| \cdot \sup |u_k^{(1)}(\cdot)|$$

$$\leq |a_k| \cdot \|\underline{W}(T)\| + c_7 \, c_6 \, c_4 \tag{3.10}$$

Since a_k must be finite for the operator to make sense, the right hand side of Eq. 3.10 is finite.

Q.E.D.

Lemma 3.3*:

Given: Assumption 1, that the system 2.1 is controllable

Then: The inverse of the first derivative operator 3.6 exists for all finite values of $\underline{\pi}$ and of η_k.

Proof: This follows from two well established results. Since the system is time invariant and controllable the controllability matrix is positive definite over any interval $[t_0, t_1]$.

$$\det \int_{t_0}^{t_1} q(\tau)q'(\tau)d\tau > 0 \quad \text{for all} \quad t_1 > t_0 \tag{3.11}$$

Since the approximate control function u_k is monotone increasing its derivative is always positive. Then

$$\{a_k + u_k^{(1)}[\underline{q}'(t)\underline{\pi}]\} > 0 \quad \text{for all finite } \underline{\pi} \text{ and for all finite time } t \tag{3.12}$$

Also, since the function 3.12 is analytic it can be approximated as closely as desired over the interval $[0, T]$ by a finite series of step functions $c_j 1(t_j)$, where the t_j form a suitable partition of $[0, T]$.

$$\left|\{a_k + u_k^{(1)}[\underline{q}'(t)\underline{\pi}]\} - \sum_{j=1}^{\ell(\epsilon)} c_j[1(t_j) - 1(t_{j-1})]\right| < \epsilon$$

The coefficients c_j can be required to satisfy

$$c_j > 0, 1 \leq j \leq \ell(\epsilon) \tag{3.13}$$

From Eqs. 3.11 and 3.13 an approximation to the first derivative is formed, which satisfies

---

*This result was pointed out by Professor Roger Brockett.

$$\det \sum_{j=1}^{\ell(\epsilon)} c_j \int_{t_{j-1}}^{t_j} \underline{q}(\tau)\underline{q}'(\tau)d\tau > 0 \qquad (3.14)$$

Equation 3.14 can be bounded away from zero as $\epsilon \to 0$, and its argument converges to the first derivative operator. Hence

$$\det T_k^{(1)}(\underline{\pi}) = \det \int_0^T \underline{q}(\tau)\underline{q}'(\tau)\{a_k + u_k^{(1)}[\underline{q}'(\tau)\underline{\pi}]\}d\tau > 0$$

and $$[T_k^{(1)}(\underline{\pi})]^{-1} \quad \text{exists}$$

Q.E.D.

There is a near singular condition if the quantity in braces in Eq. 3.6 is very small. This occurs in a difficult or impossible problem sometimes as $a \to 0$. The costate initial condition guess may already be such as to lead to the use of almost all the available control effort, thus making $u_k^{(1)}(.)$ very small. This is a near singular condition, leading to a small determinant and possible numerical difficulty in finding the inverse.

Theorem 3.1:

Assume: Assumptions 3.1 and 3.2.

Then: 1. There is a solution to the first operator equation of the sequence 2.20, i.e., to the linear equation

$$T_0(\underline{\pi}) = \underline{\xi} - a_0 \underline{W}(T)\underline{\pi} = \underline{0} \qquad (3.15)$$

2. Starting with the general member of the sequence 2.20 and its solution

$$T_k(\underline{\pi}_k) = \underline{0}$$

using a_k and η_k (3.16)

There exist changes $\Delta\eta_k > 0$ and $\Delta a_k < 0$ (or $\Delta a_k = 0$ if $a_k = 0$) such that Newton's method converges when applied sequentially (see Definition 2.2) to the operator equation

$$T_{k+1}(\underline{\pi}) = 0 \quad \text{where} \quad \begin{aligned} a_{k+1} &= a_k + \Delta a_k \\ \eta_{k+1} &= \eta_k + \Delta\eta_k \end{aligned} \tag{3.17}$$

Proof: 1. The linear equations 3.15 are solved by inspection

$$\underline{\pi}_0 = 1/a_0 \, \underline{W}^{-1}(T)\underline{\xi} \tag{3.18}$$

and the controllability matrix possesses an inverse by Assumption 3.1.

2. Theorem B.1 from Appendix B is to be applied.

Existence of the inverse of the first derivative operator implies that each element of the matrix $[T_k^{(1)}(\underline{\pi})]^{-1}$ is finite. This in turn means that the matrix has finite norm, and at any particular argument $\underline{\pi}_\ell$ this norm can be bounded from above by a constant.

$$\| [T_k^{(1)}(\underline{\pi}_\ell)]^{-1} \| \leq c_{\underline{\pi}_\ell} \tag{3.19}$$

The main task is to show that given any $\epsilon > 0$, there exists some $\delta > 0$ such that setting

$$\begin{aligned} a_{k+1} &= \max\{0, a_k - \delta\} \\ \eta_{k+1} &= \eta_k + \delta \end{aligned} \tag{3.20}$$

leads to $$\| T_{k+1}(\underline{\pi}_k) \| < \epsilon \tag{3.21}$$

By definition, $$\| T_k(\underline{\pi}_k) \| = 0 \tag{3.22}$$

Property 3.21 can then be shown by examining the operator T_{k+1}.

$$\begin{aligned} \| T_{k+1}(\underline{\pi}_k) \| &= \| T_{k+1}(\underline{\pi}_k) - T_k(\underline{\pi}_k) + T_k(\underline{\pi}_k) \| \\ &\leq \| T_{k+1}(\underline{\pi}_k) - T_k(\underline{\pi}_k) \| + \| T_k(\underline{\pi}_k) \| \end{aligned}$$

so $$\| T_{k+1}(\underline{\pi}_k) \| \leq \| T_{k+1}(\underline{\pi}_k) - T_k(\underline{\pi}_k) \| \tag{3.23}$$

Substitution of operator 2.19 into 3.23 yields

$$\| T_{k+1}(\underline{\pi}_k) \| \leq \| (a_{k+1} - a_k)\underline{W}(T)\underline{\pi}_k - \int_0^T \underline{q}(\tau)\{u_{k+1}[\underline{q}'(\tau)\underline{\pi}_k]$$

$$- u_k[\underline{q}'(\tau)\underline{\pi}_k]\}d\tau \|$$

and as a bound,

$$\| T_{k+1}(\underline{\pi}_k) \| \leq |a_{k+1} - a_k| \cdot \| \underline{W}(T) \| \cdot \| \underline{\pi}_k \|$$

$$+ \| \int_0^T |\underline{q}(\tau)| d\tau \| \cdot \sup [u_{k+1}(W) - u_k(W)] \tag{3.24}$$

For any vector $\underline{\pi}_k$ <u>of finite norm</u>, the first term can be made as small as desired by reducing δ. The approximate control function $u_k(\cdot)$ viewed as a function of η_k is continuous in the supremum norm. That is, a small change in η_k results in only a small change in $u_k(\cdot)$ at <u>any</u> value of its argument. Thus the second term in Eq. 3.24 can be made as small as desired by reducing δ. This proves property 3.21.

Part 2 of Theorem 3.1 follows from Eq. 3.19, Eq. 3.21, and Eq. 3.9. Theorem B.1 from Appendix B guarantees the convergence of Newton's method whenever

$$\| T^{(2)}_{k+1}(\underline{\pi}) \| \cdot \| [T^{(1)}_{k+1}(\underline{\pi}_k)]^{-1} \|^2 \cdot \| T_{k+1}(\underline{\pi}_k) \| < 1/2 \tag{3.25}$$

Substituting into 3.25 the constants defined earlier leads to the value of ϵ that is needed for Eq. 3.21.

$$\| T_{k+1}(\underline{\pi}_k) \| < (2c_5 c_7 c_6^2 c_{\underline{\pi}_k}^2)^{-1} \tag{3.26}$$

Constants c_5, c_6, and c_7 are fixed by the plant and the time interval. Once $\underline{\pi}_k$ is known, $c_{\underline{\pi}_k}$ is fixed. If any of these constants were

zero, Eq. 3.25 would be satisfied automatically. Otherwise δ can clearly be chosen small enough to satisfy Eq. 3.26 and thus guarantee convergence by Theorem B.1.

Q.E.D.

Comment 3.2: In part 2 of the theorem, Eq. 3.21 can be shown in another way. Take the derivative of the operator $T_{k+1}(\underline{\pi})$ with respect to α_k and η_k. These derivatives can be shown to exist and to be finite in a neighborhood around $\underline{\pi}_k$. Equation 3.21 then follows.

Comment 3.3: Equation 3.24 predicts the behavior which is encountered when Assumption 3.2 is not satisfied (an impossible problem--not in the set of reachable states at time T). In this case the sequence can still be started, but as α_k is decreased toward zero $\|\underline{\pi}_k\|$ increases without bound so that the first term of Eq. 3.24 cannot be made smaller than a certain number, i.e., the norm of a vector large enough to place the initial state $\underline{\xi}$ in the set of reachable states at time T. Thus one would never reach a step in the sequence where δ could be taken large enough to make α_{k+1} zero and still have convergence of Newton's method.

3. Convergence to the Original Operator $T(\underline{\pi})$

It remains to show that the sequence of approximate operators $\{T_k(\underline{\pi})\}$ can be made to converge to the exact operator $T(\underline{\pi})$. It is relatively easy to show that an operator $T_k(\underline{\pi})$ can be found to approximate $T(\underline{\pi})$ as closely as desired.

Theorem 3.2:

Given: The operators 2.14 and 2.19
Assumptions 3.1, 3.2, and 3.3

Then: For any $\epsilon > 0$ there exists a number $\eta(\epsilon)$ such that for all

$$\eta_k > \eta(\epsilon) \tag{3.27}$$

and

$$\alpha_k = 0$$

$$\|T(\underline{\pi}^*) - T_k(\underline{\pi}^*)\| < \epsilon \tag{3.28}$$

Proof: Assumption 3.1 and 3.3 guarantee that the problem is normal and that the argument of the control, $\underline{q}'(t)\underline{\pi}^*$, does not remain constant for any finite time interval (for a proof of this see Athans and Falb,[4] pp. 443-447).

Let condition $|\underline{q}'(t)\underline{\pi}^*| = 1$ occur m times at times t_i, i.e.,

$$|\underline{q}'(t_i)\underline{\pi}^*| = 1 \quad i = 1, 2, \ldots m \tag{3.29}$$

where $t_i \in [0, T]$

Since $\underline{q}'(t)\underline{\pi}^*$ is a continuous function which is never constant, each of the times t_i must be separated from its neighbors by some finite amount. Hence m is finite; i.e.,

$$|t_{i+1} - t_i| > 0 \quad i = 1, 2, \ldots m$$

$$\therefore \; m < \infty$$

Now proceed by removing a small time interval from $[0, T]$ around each of these m points. Let t_{i-} and t_{i+} be the end points of the i^{th} such interval and let β_i denote the interval. Let B denote the set of all such intervals.

$$B = \{t: t \in [t_{i-}, t_{i+}], \; i = 1, 2, \ldots m\} \tag{3.30}$$

The end points $t_{i\pm}$ are to be chosen such that

$$|\underline{q}'(t_{i\pm})\underline{\pi}^*| - 1 = \pm 1/\sqrt{\eta} \tag{3.31}$$

Subject to the condition

$$B \subset [0, T]$$

Note: $t_{i\pm}$ means either t_{i+} or t_{i-}. A typical division is shown in Fig. 3.1.

For small values of η two or more of the intervals may overlap. As η increases the continuity of $\underline{q}'(t)\underline{\pi}^*$ guarantees that for some finite value of η all the intervals will be separate.

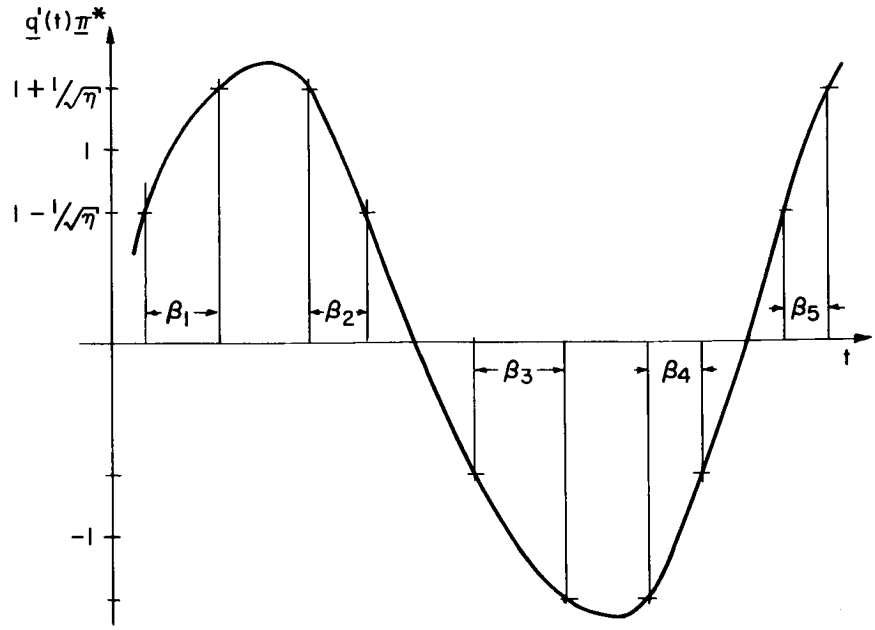


Fig. 3.1 Partition of the Time Interval for Theorem 3.2

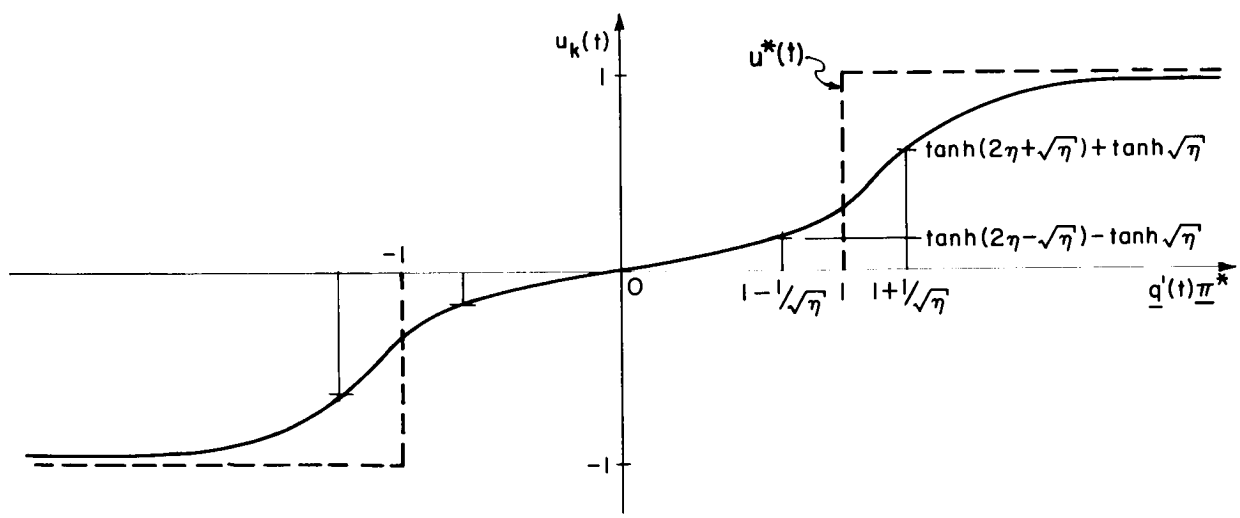


Fig. 3.2 Partition of the Approximate Control u_k for Theorem 3.2

The point is that by subdividing the time interval $[0, T]$, the difference between the exact and the approximate operators at $\underline{\pi}^*$ can be bounded.

$$T(\underline{\pi}^*) - T_k(\underline{\pi}^*) = -\int_0^T \underline{q}(\tau)\{\mathrm{dez}[\underline{q}'(\tau)\underline{\pi}^*] - u_k[\underline{q}'(\tau)\underline{\pi}^*]\}d\tau \qquad (3.32)$$

Finally, to bound the term in braces above. Figure 3.2 shows that the difference between the $\mathrm{dez}[t]$ and $u_k[t]$ functions for a given η_k increases as the ± 1 points are approached. So outside the set B the errors increase toward the times $t_{i\pm}$, and are largest where $|\underline{q}'(t)\underline{\pi}^*| = 1 \pm 1/\sqrt{\eta}$. Inside the set B it is accurate enough to bound the difference by 1.0.

Splitting the integral in operator 3.32, taking the norm and simplifying yields,

$$T(\underline{\pi}^*) - T_k(\underline{\pi}^*) \leq \int_B |q(\tau)|d\tau \cdot 1$$

$$+ \int_{[0,T]-B} |q(\tau)|d\tau[1 - \tfrac{1}{2}\tanh(2\eta+\sqrt{\eta}) - \tfrac{1}{2}\tanh\sqrt{\eta}]$$

$$\|T(\underline{\pi}^*) - T_k(\underline{\pi}^*)\| \leq \|\int_B |q(\tau)|d\tau + \int_{[0,T]-B} |q(\tau)|d\tau[1-\tanh\sqrt{\eta}]\| \qquad (3.33)$$

or by regrouping terms,

$$\|T(\underline{\pi}^*) - T_k(\underline{\pi}^*)\| \leq \|c_7[1-\tanh\sqrt{\eta}] + \int_B |\underline{q}(\tau)|d\tau \cdot \tanh\sqrt{\eta}\| \qquad (3.34)$$

Since $$\lim_{\eta \to \infty} \tanh\sqrt{\eta} = 1$$

and $$\lim_{\eta \to \infty} \quad B = \sum_{i=1}^{m} t_i \quad \text{(i.e., of measure zero)}$$

Then both terms of Eq. 3.34 can be made as small as desired by increasing η. This completes the proof.

The remaining important step is to ask how many members of the sequence will be required to reach a close approximation to the exact operator $T(\underline{\pi})$. Theorem 3.1 shows that finite steps can always be taken. One would like to reach engineering accuracy using relatively few approximate operators $T_k(\underline{\pi})$, say not more than ten or fifteen. Practical experience with Problem 1 indicates that when α can be driven to zero this objective is met, as indicated in Chapter VI. Unfortunately, there is no available proof of this property. The main reason for this is that as η_k increases, the approximate control u_k becomes more uneven, causing Theorem B.1 to give an increasingly conservative estimate of the guaranteed step size available in η.

It seems reasonable that if $T_k(\underline{\pi})$ is close to $T(\underline{\pi})$ as in Theorem 3.2 their solutions $\underline{\pi}_k$ and $\underline{\pi}^*$ will usually be close also. A proof of this follows.

Theorem 3.3

Given: 1. The operator equations 2.14 and 2.19 of Problem 1

2. Assumptions 3.1, 3.2, 3.3.

Then: For any $\epsilon_1 > 0$ there exists a number $\eta(\epsilon)$ such that,

whenever $$\eta_k > \eta(\epsilon) \tag{3.35}$$

then $$\|\underline{\pi}^* - \underline{\pi}_k\| < \epsilon_1 \tag{3.36}$$

Note: for convenience the operator equations are written below.

The exact operator

$$T(\underline{\pi}^*) = \underline{\xi} - \int_0^T \underline{q}(\tau)\, \mathrm{dez}\, [\underline{q}'(\tau)\underline{\pi}^*]\, d\tau = \underline{0} \tag{2.14}$$

The approximate operator

$$T_k(\underline{\pi}_k) = \underline{\xi} - \int_0^T \underline{g}(\tau) u_k[\underline{g}'(\tau)\underline{\pi}_k]\, d\tau = \underline{0} \tag{2.19}$$

Proof: 1. The approximate operator 2.19 has a nonsingular first derivative from Lemma 3.3. Then the inverse function theorem of analysis (see e.g., Dieudonne[25]) guarantees the existence of an inverse operator to T_k around the point $\underline{\pi}^*$. Define

$$\underline{\gamma} = T_k(\underline{\pi}^*) \tag{3.37}$$

More precisely, there exists two open sets X and Y such that

$$\underline{\pi}^* \in X \quad \text{and} \quad \underline{\gamma} \in Y$$

$$Y = T_k(X)$$

T_k^{-1} is defined on Y such that

$$T_k^{-1}(Y) = X$$

$$T_k^{-1} \in C_1 \text{ on } Y \text{ (is differentiable)}$$

and finally

$$T_k^{-1}(T_k(\underline{\pi})) = \underline{\pi} \text{ for all } \underline{\pi} \in X$$

2. From Assumptions 3.1 and 3.3 the exact operator $T(\underline{\pi})$ is one to one in a neighborhood of $\underline{\pi}^*$. (The neighborhood is assured only if $T > T^*$.) Then one can write

$$\underline{\pi}^* = T^{-1}(\underline{0}) \tag{3.38}$$

Note: More generally the operator $T(\underline{\pi})$ has an inverse in a neighborhood of the point $\underline{0}$.

3. By Theorem 3.2, the point $\underline{\gamma} = T_k(\underline{\pi}^*)$ can be brought as close as desired to the point $\underline{0} = T(\underline{\pi}^*)$. Thus by taking η large enough the open set Y can be made to include the point $\underline{0}$. Under this condition one can bring in the solution to the approximate operator equation

$$\underline{\pi}_k = T_k^{-1}(\underline{0}) \tag{3.39}$$

This situation is shown in Fig. 3.3

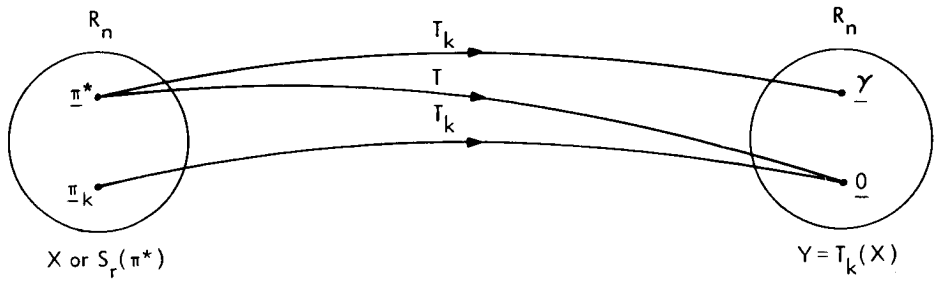


Fig. 3.3 Sets Used in Theorem 3.3

4. Since T_k^{-1} is differentiable it is also continuous. Thus if

$$\| \underline{0} - \underline{\gamma} \| < \epsilon_2 \tag{3.40}$$

then

$$\| \underline{\pi}^* - \underline{\pi}_k \| < c_8(\eta) \cdot \epsilon_2 \tag{3.41}$$

for small ϵ_2

Where the constant c_8 in general depends on how large η_k is. Under the given assumptions, the exact operator $T(\underline{\pi})$ can also be shown to be continuous near $\underline{\pi}^*$. So given some lower bound $\eta_{\ell 0}$ on η_k such that $\eta_k > \eta_{\ell 0}$, an upper bound on $c_8(\eta)$ can be found. Then Eq. 3.41 can be rewritten.

$$\| \underline{\pi}^* - \underline{\pi}_k \| < c_8 \cdot \epsilon_2 \tag{3.42}$$

5. The rest is simple arithmetic. Given ϵ_1 in Eq. 3.36, one must make $c_8 \cdot \epsilon_2$ in Eq. 3.42 at least as small by requiring

$$\epsilon_2 \leq \epsilon_1 / c_8 \tag{3.43}$$

But Theorem 3.2 guarantees Eq. 3.43 if η is large enough. η_k must thus be chosen large enough to both validate Eq. 3.39 and satisfy Eq. 3.43. For any nonzero value of ϵ_1 in Eq. 3.36 this can be done with a finite value of η_k. This completes the proof.

The new results pertaining to Newton's method were introduced in showing the rationale behind the sequence of approximate operators. General characteristics of Newton's method are shown in Appendix B. Quadratic convergence is a very nice property to have.

In conclusion, some factors affecting the performance of the method have been brought out. It was shown for the class of problems treated that Newton's method can always be carried out. A sequence of approximate operators can be guaranteed such that Newton's method converges when applied to each one in turn. This sequence approaches the exact operator, and if an optimal solution exists, it converges to it. The suboptimal controls defined by the approximate operators do not require the rapid switching of the optimal control, and from a theoretical point of view should be efficient in their use of fuel. These properties are confirmed experimentally in Chapter V.

CHAPTER IV

ORGANIZATION OF COMPUTER PROGRAM

In order to test the procedure outlined in Chapter II, a computer program was written in the Fortran II language. In this chapter the philosophy behind the program and the organization of its parts are explained. The overall structure and methods are described first; then each subroutine is described in greater detail.

A listing of the program is found in Appendix D.

A. OVERALL STRUCTURE AND PHILOSOPHY

It was desired to have a program that would be as flexible as possible within the framework of Problem 1, and at the same time, as easy to use and as comprehensive as possible. For instance, the fundamental matrix is computed directly from its series definition, so that any system matrix $\underline{A}$ can be used.

Certain parameters have been left available for adjustment--they will not normally be changed, but can be used to alter the sequence at each step of Newton's method or change the sequence step size or the point beyond which the sequence terminates (how closely the exact operator $T(\underline{\pi})$ is finally approximated). Even the accuracy of computation of the fundamental matrix and the accuracy of the solution of the members $T_k(\underline{\pi})$ of the sequence can be adjusted. However, all these parameters have normal values built in, and need not be touched by the user. The normal values were determined by experiment. These parameters will be described in more detail in the sections on the subroutines in which they are used.

A flow chart is shown in Fig. 4.1 with the essential portions of the program, showing the relations which were used for the computer solution. This shows logically and in order the various steps used on the computer. For the purpose of convenience in writing and debugging,

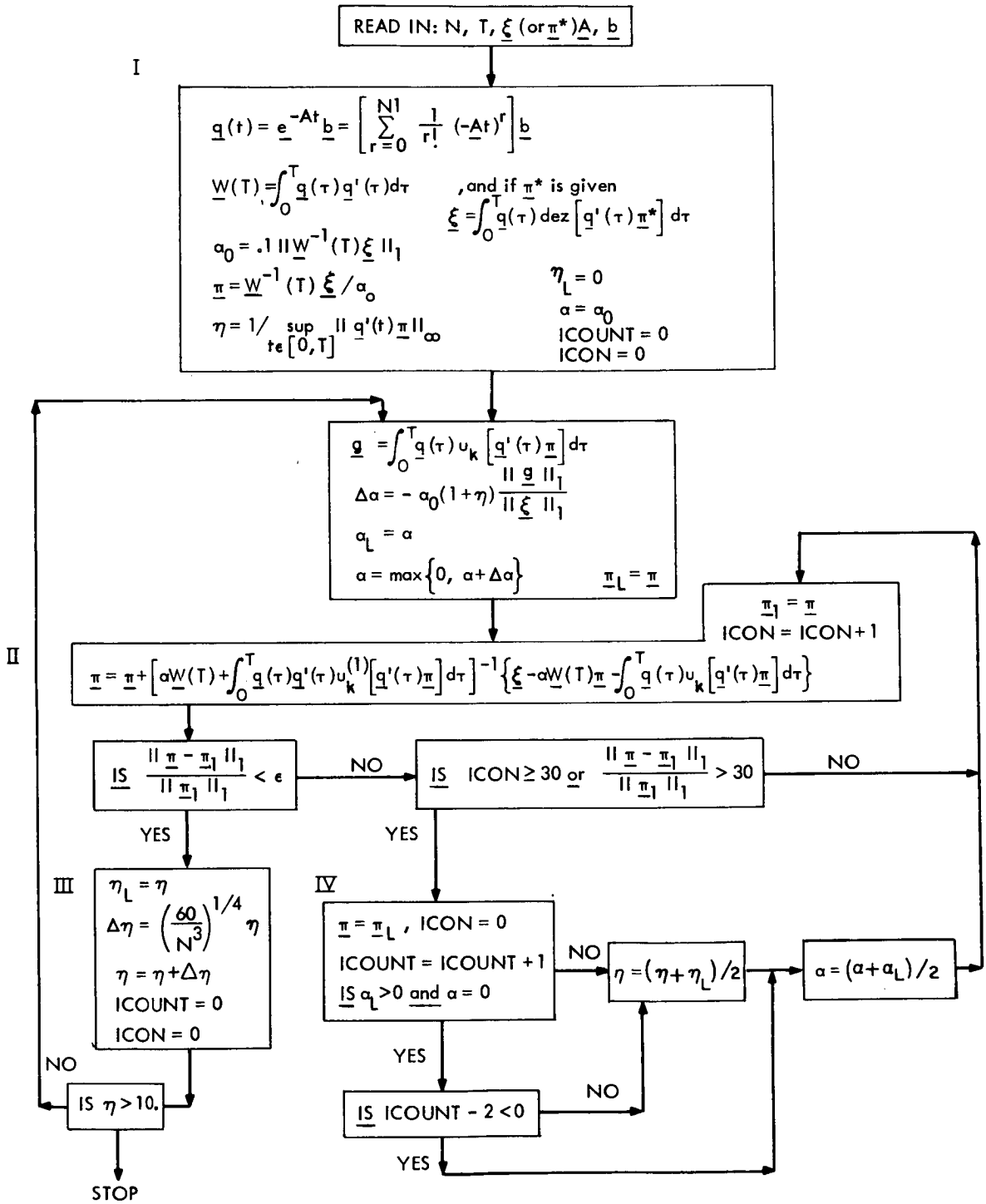


Fig. 4.1 Digital Computer Flow Chart

the program has been broken into units called subroutines. There is a Main program (actually a subroutine like the others) which assumes most of the readin-printout and the internal routing responsibilities. The various other subroutines are connected to the Main program and operate in more or less of a sequence. This sequence corresponds to the flow diagram, so that the (essential) subroutines correspond to certain areas of the flow diagram. A chart showing the various subroutines is included as Fig. 4.2.*

To evaluate integrals by numerical means, a simple trapezoidal rule has been used. In order to simplify the writeup, the symbol $\sum_{\ell=1}^{M}$ will be used as the numerical approximation for $\int_0^T$. That is, the expression $\sum_{\ell=1}^{M} S_\ell$ will really mean $\sum_{\ell=2}^{M} S_\ell + \frac{1}{2} S_1 + \frac{1}{2} S_{M+1}$.

B. MAIN PROGRAM (SUBROUTINE MAIN)

This subroutine has as its primary purpose the control of the flow of computation. After each other subroutine is finished, the computer returns to subroutine MAIN to decide which one to go to next. Because of this, there is a single place where one can look to follow the overall course of computations.

Decisions as to which optional subroutines to use are made in subroutine MAIN. The variable ICHO is used for this purpose. Figure 4.3 shows a diagram of possible values for this control variable and their meanings. Note that the user has separate control over whether the subroutines CKCON and SSTRAJ are used.

All of the data is read in by subroutine MAIN. The required format is indicated in Appendix D. The data include the variables n, T, $\underline{\xi}$, $\underline{\theta}$, $\underline{A}$, $\underline{b}$, and the decision constants. The decision constants are

---

* Several subroutines have been added which are not essential but either are convenient or else make the operation more complete. These are shown in Fig. 4.2 with dotted lines connecting them to the essential portions of the program.

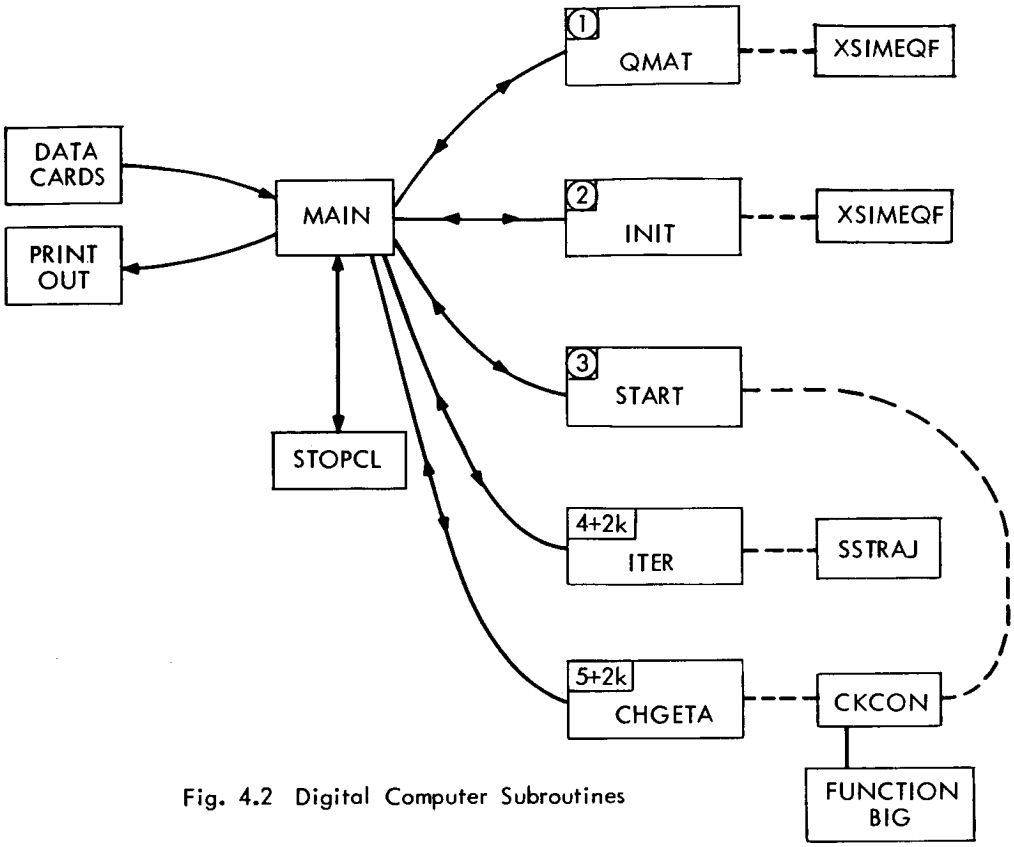


Fig. 4.2 Digital Computer Subroutines

ICHO	OPTIONAL SUBROUTINES USED			
0				
1				
2				use SSTRAJ
3		use CKCON	Require h<1/2	
4				
5				use SSTRAJ
6				
7				
8				use SSTRAJ
9				
10	Compute $\underline{\xi}$ from $\underline{\pi}^*$			
11				
12				use SSTRAJ
13		use CKCON	Require h<1/2	
14				
15				use SSTRAJ
16				
17				
18				use SSTRAJ

Fig. 4.3 The Control Constant ICHO

named EPS, AMAX, EPMTX, ALPT, M, ICHO, and KPETA. Their roles will be discussed in the subroutines where they are used.

Most of the output is printed out by subroutine MAIN. First of all, the input data is printed out for identification and checking. The matrix exponential and the controllability matrix are printed out. The parameters of each approximate operator are indicated, and the costate initial condition vector $\underline{\pi}^i$ is included for each step of Newton's method. If anything goes wrong, such as a matrix inversion difficulty, an apropos warning statement is given and appropriate action is taken, i.e., the program stops if this inversion becomes too difficult numerically.

In addition, the optional subroutine CKCON leads to the printing out of the convergence theorem parameter h. Some of the other subroutines also do some printing on their own, where this is noticeably more convenient. Thus, subroutine SSTRAJ prints the trajectory and control variable argument rather than transfer this information. Subroutine CKCON prints some internal information. Subroutine STOPCL is a library routine, which prints the elapsed real time.

For the output format, the usual eight decimal places with an exponent are printed, even though in many cases only two or three of these places are significant.

Finally, subroutine MAIN decides when the program should stop, by comparing the new value of η found in subroutine CHGETA with the given decision constant AMAX. The program stops when

$$\eta_k > \text{AMAX} \tag{4.1}$$

Experience shows that the procedure has converged pretty well by the time $\eta = 5$ or 10, so as a normal value one uses

$$\text{AMAX} = 10.$$

C. SUBROUTINE QMAT

This subroutine computes the matrix exponential and the $\underline{q}$ vector. The series definition of the matrix exponential is used for the computation

$$\underline{e}^{\underline{A}t} \triangleq \sum_{\ell=0}^{\infty} \frac{1}{\ell!} (\underline{A}t)^{\ell}$$

This form allows the system dynamics $\underline{A}$ to be given as any square matrix. Further, the series is guaranteed to converge for any square matrix $\underline{A}$ satisfying $\| \underline{A} \| < \infty$, and any time satisfying $t < \infty$.

A decision constant, M, is used to break the given time interval $[\ 0,T\ ]$ into m equal increments δ.

$$\delta = T/M \tag{4.2}$$

This program handles values of $M \leq 100$, but can easily be extended to higher values. The continuous problem is replaced for computation purposes by a discrete one. Hence it becomes necessary to compute $\underline{e}^{\underline{A}\delta}$. Then $\underline{e}^{\underline{A}2\delta}$, $\underline{e}^{\underline{A}3\delta}$, etc., can be found by matrix multiplication, which is much simpler than computing each of these by the series 4.1

An iterative form for computer application is then simply,

$$\underline{D}_{\ell} = \underline{D}_{\ell-1} \cdot (\underline{A}\delta)/\ell \tag{4.3}$$

and

$$\underline{E}_{\ell} = \underline{E}_{\ell-1} + \underline{D}_{\ell} \tag{4.4}$$

with the initial conditions

$$\underline{D}_0 = \underline{E}_0 = \underline{I} \tag{4.5}$$

Finally, in order to decide when to stop, the contribution of each new term is compared with the size of the series summation so far. Another decision constant EPMTX is read in with the data just for this purpose, and a simplified way of comparing sizes is used. Thus the computation stops when

$$\max_{i,j} \frac{\underline{D}_{\ell}}{\underline{E}_{\ell}} \leq \text{EPMTX}$$

Actually, in order to keep the computations to a minimum, the test is performed on each row separately. So the series for the $i^{\underline{th}}$ row can

be terminated when

$$\max_j \frac{d_{i,\ell}}{e_{i,\ell}} \leq \text{EPMTX}. \tag{4.6}$$

To keep the accuracy high, a small value was used for EPMTX

$$\text{EPMTX} = 10^{-6}$$

It is really $\underline{e}^{-\underline{A}t}$ which is used, so the matrix $\underline{e}^{\underline{A}\delta}$ is next inverted, by using the library tape function XSIMEQ.

The intermediate matrices $\underline{e}^{-\underline{A}k\delta}$ need not be stored, as they are needed only in the combination

$$\underline{q}(k) = \underline{e}^{-\underline{A}k\delta}\underline{b} \tag{4.7}$$

So once $\underline{e}^{-\underline{A}\delta}$ is known, the procedure is

$$\underline{F}_k = \underline{F}_{k-1} \cdot \underline{e}^{-\underline{A}\delta} \tag{4.8}$$

and

$$\underline{q}(k) = \underline{F}_k\underline{b} \tag{4.9}$$

$$k = 1, 2, \ldots M + 1$$

With the initial condition,

$$\underline{F}_0 = \underline{I}$$

For later use, the matrix $\underline{e}^{-\underline{A}T}$ is also stored.

$$\underline{F}_{M+1} = \underline{e}^{-\underline{A}T} \tag{4.10}$$

Finally, the supremum norm is applied to each function in the $\underline{q}(t)$ vector function. In the discrete version this means

$$q_{i,\ sup} = \sup_{k=1,\ M+1} q_i(k)$$

and the resulting vector is stored.

$$\underline{q}_{sup} = \begin{bmatrix} q_{1,sup} \\ \vdots \\ q_{n,sup} \end{bmatrix} \tag{4.11}$$

D. SUBROUTINE INIT

Here the controllability matrix and the first guess for the initial costate vector $\underline{\pi}_0$ are computed. If desired, the state initial condition vector is also found. The controllability matrix is,

$$\underline{W}(T) = \int_0^T q(\tau)q'(\tau)d\tau$$

or in the discrete version, using the trapezoidal rule and the simplified summation convention given in Section A of this chapter,

$$\underline{W}(T) = \delta \sum_{\ell=1}^{M} \underline{q}(\ell)\underline{q}'(\ell). \tag{4.12}$$

Element by element, this is

$$w_{ij}(T) = \delta \sum_{\ell=1}^{M} q_i(\ell)q_j(\ell). \tag{4.13}$$

Again, the inverse is needed. It is found by using the tape library function XSIMEQ, and is stored as WI(I, J).

If the control constant ICHO is larger than 10, the vector read in for the state equation initial condition $\underline{\xi}$ is taken to be the optimal costate initial condition $\underline{\pi}^*$, and a new state initial condition vector $\underline{\xi}$ is computed based on

$$\underline{\xi} = \int_0^T \underline{q}(\tau)dez[\underline{q}'(\tau)\underline{\pi}^*]d\tau \tag{4.14}$$

or a simple discrete form of Eq. 4.14, again using the simplified summation for the trapezoidal rule,

$$\underline{\xi} = \sum_{\ell=1}^{M} \underline{q}(\ell)\mathrm{dez}\ [\underline{q}'(\ell)\underline{\pi}^*] \tag{4.15}$$

In addition, when $\underline{\xi}$ is computed the cost (i.e., fuel) is found for the optimal control $u^*(t)$.

$$J(u^*) = \int_0^T |u^*(\tau)|\, d\tau \tag{4.16}$$

or a simple discrete form of Eq. 4.16

$$J(u^*) = \delta \sum_{\ell=1}^{M} |\mathrm{dez}[\underline{q}'(\ell)\underline{\pi}^*]| \tag{4.17}$$

The given optimal costate initial condition $\underline{\pi}^*$ is now discarded and will <u>not</u> be used in any of what follows.

The next step, whether $\underline{\xi}$ is given or computed is to adjust it to take care of the possibility that the final condition on the state vector, $\underline{\theta}$, may not be zero. In this case there is always an equivalent initial condition $\underline{\zeta}$ such that the problem of controlling from the state $\underline{\xi}$ to the state $\underline{\theta}$ in T seconds is equivalent to controlling from the state $\underline{\zeta}$ to the origin $\underline{0}$ in T seconds.

$$\underline{\zeta} = \underline{\xi} - \underline{e}^{-\underline{A}T}\underline{\theta} \tag{4.18}$$

Of course, such a transformation is valid only for linear systems, and takes no account of what happens to the state for time $t > T$ seconds.

Finally the subroutine INIT takes care of the first approximate operator $T_0(\underline{\pi})$. From Eq. 3.18 the solution is known, but the constant α_0 must be chosen. It determines the length of the vector $\underline{\pi}_0$, and is chosen to make this vector 10 units in the ℓ_1 norm. Remember that the dead-zone function has its "turn-on" magnitude normalized to 1. So the number 10 is a compromise, designed to guarantee quite a bit of control "on" time, but still leave some "coasting"

time when various different plants are given. The effect of this compromise is discussed in Chapter VI for the computer examples used.

The ℓ_1 norm is used in calculating a_0.

$$a_0 = \frac{1}{10} \| \underline{W}^{-1}(T)\underline{\zeta} \|_1 \tag{4.19}$$

then the solution to the operator $T_0(\underline{\pi})$ is, using Eq. 3.18,

$$\underline{\pi}_0 = \frac{1}{a_0} \underline{W}^{-1}(T)\underline{\zeta} = 10 \frac{\underline{W}^{-1}(T)\underline{\zeta}}{\| \underline{W}^{-1}(T)\underline{\zeta} \|_1} \tag{4.20}$$

E. SUBROUTINE START

At this point the first step away from linearity is taken by choosing the operator $T_1(\underline{\pi})$. This subroutine is similar to subroutine CHGETA except that here the first nonzero value of η is chosen.

1. Choose η_1

As discussed in Chapter II, the objective is to make η_1 large, but not so large that Newton's method will not converge. Since the form of Eq. 4.32 does not permit it to be applied where $\eta_k=0$, another approach is used. Increasing η increases the magnitude of the control function $u_k(t)$. Thus if $\Delta\eta$ is too large, the function $\eta_1 \underline{q}'(t)\underline{\pi}_0$ will result in something approaching a maximum effort controller. To be on the safe side, η_1 is chosen to limit the maximum possible value of $\eta_1 \underline{q}'(t)\underline{\pi}_0$ to one, i.e., in discrete form,

$$\max_{j=1,\ M} \eta_1 \underline{q}'(j)\underline{\pi}_0 \leq 1.$$

A simple bound for this is found by using the vector $\underline{q}_{sup}$ of Eq. 4.11.

$$\| \eta_1 \underline{q}_{sup} \underline{\pi}_0 \|_\infty = 1$$

or

$$\eta_1 = 1/\| \underline{q}_{sup} \underline{\pi}_0 \|_\infty \tag{4.21}$$

The point is that when iterations are started on the operator $T_1(\underline{\pi})$ the argument of the control variable will remain small enough so that changes in the vector $\underline{\pi}$ will have a large effect ($\frac{\partial T_1(\underline{\pi})}{\partial \underline{\pi}}$ has relatively large elements). This is admittedly a rough approximation, but it gets the sequence of operators under way.

2. Choose α_1

Now that η_1 is chosen, α_1 is found just as in subroutine CHGETA. The method is discussed in Section D of Chapter II.

a. Estimate how effective the control will be

$$\underline{g}_1 = \int_0^T \underline{q}(\tau) u_1[\underline{q}'(\tau)\underline{\pi}_0]\, d\tau \qquad (4.22)$$

b. Decide how much slope should be removed.

$$\Delta\alpha = -\alpha_0(1 + \eta_1)\frac{\|\underline{g}_1\|_1}{\|\underline{\zeta}\|_1} \qquad (4.23)$$

c. Take the slope α_1 as near zero as this allows

$$\alpha_1 = \max\{0, \alpha_0 + \Delta\alpha\} \qquad (4.24)$$

F. SUBROUTINE ITER

This is the key subroutine, in which Newton's method is applied to the operator $T_k(\underline{\pi})$, using $\underline{\pi}_{k-1}$ as a starting approximation.

There are three values of the vector $\underline{\pi}$ which are of importance

$\underline{\pi}^i$ is the current iterate

$\underline{\pi}_\ell$ is the previous iterate, $\underline{\pi}_\ell = \underline{\pi}^{i-1}$

$\underline{\pi}_{ost} = \underline{\pi}_{k-1}$ is the starting approximation

The first step is to store the starting approximation

$$\underline{\pi}_{ost} = \underline{\pi}_{k-1}$$

also, at each step, the previous value of $\underline{\pi}$ is stored.

$$\underline{\pi}_{\ell} = \underline{\pi}^{i}$$

Then a step of Newton's method is taken, as discussed in Section E of Chapter II.

$$\underline{\pi}^{i+1} = \underline{\pi}^{i} + \left[\alpha_k \underline{W}(T) + \int_0^T \underline{q}(\tau)\underline{q}'(\tau)u_k^{(1)}[\underline{q}'(\tau)\underline{\pi}^{i}]d\tau\right]^{-1}$$
$$\left[\zeta - \alpha_k \underline{W}(T)\underline{\pi}^{i} - \int_0^T \underline{q}(\tau)u_k[\underline{q}'(\tau)\underline{\pi}^{i}]d\tau\right] \qquad (4.25)$$

The functions u_k and $u_k^{(1)}$ are written out as Eqs. 2.23 and 2.24. At the same time, the amount of fuel used is computed

$$J(u_k) = \delta \sum_{\ell=1}^{M} |u_k[\underline{q}'(\ell)\underline{\pi}^{i}]| \qquad (4.26)$$

Now a check is made for convergence or divergence at each step using an estimate of the proportional change between $\underline{\pi}^{i+1}$ and $\underline{\pi}^{i}$.

$$\text{ERROR} = \frac{\|\underline{\pi}^{i+1} - \underline{\pi}_{\ell}\|_1}{\|\underline{\pi}_{\ell}\|_1} \qquad (4.27)$$

One of the control constants, EPS, is used to check for convergence.

If

$$\text{ERROR} \leq \text{EPS} \qquad (4.28)$$

then Newton's method converges, $\underline{\pi}^{i} = \underline{\pi}_k$, and Eq. 4.26 gives the control cost for the operator $T_k(\underline{\pi})$. A variable named ICON is set equal to 1 to indicate the convergence.

A nominal value of EPS was chosen to give engineering accuracy in the converged $\underline{\pi}_k$

$$EPS = 10^{-3}$$

There are two tests for divergence of the method. First, a count is kept of the number of iterations under the name ICON. Divergence is defined to occur if either

$$ICON \geq 30 \tag{4.29}$$

or

$$ERROR \geq 30 \tag{4.30}$$

Under this condition the present operator $T_k(\underline{\pi})$ is discarded, and the vector $\underline{\pi}_{k-1}$ is restored as the most recent solution vector.

$$\underline{\pi}_{k-1} = \underline{\pi}_{ost} \tag{4.31}$$

Also the variable ICON is set equal to 0 to indicate divergence.

Finally, if there is neither convergence or divergence at the $i+1^{st}$ step, the program goes back to Eq. 4.25 for the next iteration.

G. SUBROUTINE CHGETA

This is the last of the essential subroutines. Its purpose is to choose the next operator. There are two cases, depending on whether the last operator led to convergence or divergence in subroutine ITER.

1. When the k^{th} operator has been solved, the normal method of selecting the next operator $T_{k+1}(\underline{\pi})$ is;

a. To choose η_{k+1}, use the formula discussed in Chapter II Section D.

$$\eta_{k+1} = \eta_k + \left(\frac{60}{n^3}\right)^{1/4} \eta_k \tag{4.32}$$

b. To choose α_{k+1}, as in Section 2.D, first estimate how effective the control will be,

$$\underline{g}_k = \int_0^T \underline{q}(\tau) u_{k+1}[\underline{q}'(\tau)\underline{\pi}_k]\, d\tau$$

Then decide how much slope should be removed.

$$\Delta a_k = -a_0(1+\eta_{k+1}) \frac{\| g_k \|_1}{\| \zeta \|_1} \tag{4.33}$$

Finally, take the slope a_{k+1} as near zero as this allows.

$$a_{k+1} = \max \{0, a_k + \Delta a_k\}$$

A counting index called ICOUNT is set to zero for use in Subsection G.2 of this chapter.

2. If the solution to the $k^{\underline{th}}$ operator was not found, due to the failure of Newton's method to converge, then this $k^{\underline{th}}$ operator is discarded and a more conservative choice is made. The basic idea is to go halfway back to the $k-1^{\underline{st}}$ operator for the new $k^{\underline{th}}$ operator. Then Theorem 3.1 guarantees that a $k^{\underline{th}}$ operator will eventually be found for which Newton's method will converge to the solution.

In the computer sense in which an equation implies a replacement, the equations are,

$$\eta_k = \frac{1}{2}(\eta_k + \eta_{k-1}) \tag{4.34}$$

$$a_k = \frac{1}{2}(a_k + a_{k-1}) \tag{4.35}$$

One complication arises in trying to drive a to zero. In a difficult problem (the final state is difficult to reach), it was found to take more steps to get a to zero. But the process is made easier if η is kept at the old larger value. As a compromise, the following is done:

> If the program tried to reduce a to zero at the $k^{\underline{th}}$ step and failed (no convergence), then for the next two tries η_k is left unchanged.

In computer language this is done by counting the number of tries to get a workable $k^{\underline{th}}$ operator using the variable ICOUNT. Thus at each try,

$$\text{ICOUNT} = \text{ICOUNT} + 1$$

now if,

$$a_{k,1} = 0$$

where

$a_{k,1}$ is the first try for the $k^{\underline{th}}$ operator

and

$$a_{k-1} \neq 0$$

then Eq. 4.34 is skipped for two tries. That is, $a_{k,4}$ corresponds to $\eta_{k,2}$, etc.

Comment 4.1: The above section makes it possible for η to become large without a being reduced to zero. If the problem attempted is an impossible one, this will show up as an a that never reaches zero. It is conceivable that a problem which is "almost impossible" (final state very close to the boundary of the set of reachable states) would also result in a nonzero final value of a. Equation 4.1, the stopping condition, now provides a logical test of when to stop trying to get a to zero.

Comment 4.2: For research purposes, two decision constants KPETA and ALPT were included in the program. Normal values for them are,

$$\begin{aligned} \text{KPETA} &= 2 \\ \text{ALPT} &= 1.0 \end{aligned} \qquad (4.36)$$

However, if a change in the characteristics of the subroutine is desired, these can be changed. KPETA is the number of times η_k is retained at the value $\eta_{k,1}$ as noted above. ALPT adjusts the rate at which the linear slope a is reduced, according to the formula below (this is the real Eq. 4.33:

$$\Delta a_k = -a_0(1 + \eta_{k+1}) \frac{\| g_k \|_1}{\| \underline{\zeta} \|_1} \cdot \text{ALPT} \qquad (4.37)$$

Optional Subroutines

H. SUBROUTINE SSTRAJ

This rather simple subroutine computes the argument of the control function and the trajectory in the state space.

The argument of the control function is

$$ARG(t) = \underline{b}'\underline{e}^{-A't}\underline{\pi}_k = \underline{q}'(t)\underline{\pi}_k$$

and the trajectory in the state space is,

$$\underline{x}(t) = \underline{e}^{At}\left[\underline{\xi} + \int_0^t \underline{q}(\tau)u_k\left[ARG(\tau)\right]d\tau\right]$$

For computation, approximate discrete equations are used.

$$ARG(I) = \underline{q}'(I)\underline{\pi}_k \tag{4.38}$$

and

$$\underline{x}(I+1) = \underline{e}^{A\delta}\left[\underline{x}(I) + \tfrac{1}{2}\,\delta\underline{b}u_k\left[ARG(I)\right]\right] + \tfrac{1}{2}\,\delta\underline{b}u_k\left[ARG(I+1)\right] \tag{4.39}$$

The argument of the control function is printed out first, followed by the state vector for the same (time) index I.

I. SUBROUTINE CKCON

The sufficient condition for convergence presented in Appendix B is used in order to compute the parameter h. This allows a study of how large h can be and still have convergence take place. Also it was felt at the initial stages of this research that this information might be a guide in choosing the sequence of operators $T_k(\underline{\pi})$.

The required analytic expressions are shown in Sec. F, Ch. II. As above, the integrals are approximated numerically using the trapezoidal rule. A fortran function statement called BIG is used to find $u^{(2)}_{max}$, the maximum value of $u^{(2)}_{k+1}(\cdot)$. The expressions evaluated by the computer are;

the first derivative inverse Γ_k,

$$\gamma_{ij} = -\left[a_{k+1} w_{ij}(T) + \delta \sum_{\ell=1}^{M} q_i(\ell) q_j(\ell) u_{k+1}^{(1)} [\underline{q}'(\ell)\underline{\pi}_k]\right]^{-1} \tag{4.40}$$

and the norms,

$$A_{norm} = \max_i \left| \sum_{j=1}^{n} \gamma_{ij} \left[\xi_j - a_{k+1} \cdot \sum_{h=1}^{n} w_{jh}(T)\pi_h - \delta \sum_{\ell=1}^{M} q_j(\ell) u_{k+1} [q'(\ell)\underline{\pi}_k]\right]\right| \tag{4.41}$$

$$B_{norm} = \max_i \sum_{j=1}^{n} \sum_{k=1}^{n} \left| \sum_{h=1}^{n} \gamma_{ih} \cdot \sum_{\ell=1}^{M} q_h(\ell) \cdot q_j(\ell) \cdot q_k(\ell)\right| \cdot \text{BIG } F(u_{k+1}^{(2)}) \tag{4.42}$$

$$C_{norm} = \max_i \sum_{j=1}^{n} \sum_{k=1}^{n} \left| \sum_{h=1}^{n} \gamma_{ih} \cdot \sum_{\ell=1}^{M} q_h(\ell) \cdot q_j(\ell) \cdot q_k(\ell) \cdot u_{k+1}^{(2)} [\underline{q}'(\ell)\underline{\pi}_k]\right| \tag{4.43}$$

Then the quantities

$$h = A_{norm} \cdot B_{norm} \tag{4.44}$$

and

$$h_1 = A_{norm} \cdot C_{norm} \tag{4.45}$$

are formed to check for guaranteed and estimated convergence.

J. FUNCTION BIG

This part of the program is a function instead of a subroutine. It performs the relatively minor task of finding the maximum value of $u_{k+1}^{(2)}(\cdot)$. Actually any of the subroutines which have no more than one (scalar) variable in the argument (calling sequence) could have been made into functions. However, this part is really subordinate to the subroutine CKCON and deserves the lower rank.

Newton's method is used to search for the maximum value, using a good starting approximation.

Let

$$X = \tanh(\eta_k \underline{q}'\underline{\pi} - \eta_k)$$

$$Y = \tanh(\eta_k \underline{q}'\underline{\pi} + \eta_k)$$

$$TE = \tanh(2\eta_k)$$

then the hyperbolic functions satisfy these relations

$$Y = \frac{X+TE}{1+X\cdot TE} \tag{4.46}$$

$$\frac{dX}{d(\underline{q}'\underline{\pi})} = \eta_k(1-X^2)$$

$$\frac{dY}{d(\underline{q}'\underline{\pi})} = \eta_k(1-Y^2)$$

so the approximate control u_k and its derivative are given by,

$$u_k = \frac{1}{2}[X+Y]$$

$$u_k^{(1)} = \frac{1}{2}\eta_k[1-X^2+1-Y^2]$$

$$u_k^{(2)} = \eta_k^2[X-X^3+Y-Y^3]$$

$$u_k^{(3)} = -\eta_k^3[(1-X^2)(1-3X^2)+(1-Y^2)(1-3Y^2)]$$

In order to find an extremum for $u_k^{(2)}$, the next higher derivative is set equal to zero. Define

$$g(X,Y) = (1-X^2)(1-3X^2) + (1-Y^2)(1-3Y^2) = 0 \tag{4.47}$$

using equation 4.46 this is reduced to a function of X alone.

$$g(X) = (1-X^2)(1-3X^2) + \left[1-\left(\frac{X+TE}{1+X\cdot TE}\right)^2\right]\cdot\left[1-3\left(\frac{X+TE}{1+X\cdot TE}\right)^2\right] = 0 \tag{4.48}$$

Finally, to apply Newton's method, the derivative of Eq. 4.48 is needed.

$$\frac{dg(X)}{dX} = -4X(2-3X^2)-4\left(\frac{X+TE}{1+X\cdot TE}\right)\left[2-3\left(\frac{X+TE}{1+X\cdot TE}\right)^2\right]\frac{1-TE^2}{(1+X\cdot TE)^2}$$

or using Eq. 4.46,

$$\frac{dg(X)}{dX} = -4X(2-3X^2)-4Y\left[2-3Y^2\right]\frac{1-TE^2}{(1+X\cdot TE)^2}$$

As an empirical starting approximation, take

$$X_o = \begin{cases} 1/\sqrt{3}-.62\eta+2\eta^3 & \eta \leq .28 \\ 1/\sqrt{3}(1-e^{-5\eta}) & \eta > .28 \end{cases} \tag{4.49}$$

Then the recursive relation for Newton's method is, using subscripted i for convenience,

$$Y_i = \frac{X_i + TE}{1+X_i\cdot TE} \tag{4.50}$$

and

$$X_{i+1} = X_i + \frac{(1-X_i^2)(1-3X_i^2)+(1-Y_i^2)(1-3Y_i^2)}{4\left[X_i(2-3X_i^2)+Y_i(2-3Y_i^2)(1-TE^2)/(1+X_i\cdot TE)^2\right]} \tag{4.51}$$

Equations 4.50 and 4.51 are to be repeated until there is neglible change in X. In practice six iterations were used, although three iterations were found to be sufficient. Once the iterations are finished, the maximum value of $u_k^{(2)}(\cdot)$ is given by

$$BIG = \eta_k^2(X-X^3+Y-Y^3) \tag{4.52}$$

K. SUBROUTINE MITMR

This is a library subroutine used for measuring and recording real time by using the IBM Interval Timer Clock. It is described in the MIT Computation Center bulletin number CC-193-2. Only two com-

mands have been used from this subroutine.

1. CALL RSCLCK-causes the clock to be set to zero.
2. CALL STOPCL(I)-gives the elapsed time from the last clock reset, in $60^{\underline{ths}}$ of a second.

L. SUBROUTINE XSIMEQF

This library subroutine solves the matrix equation

$$\underline{P}\,\underline{X} = \underline{Q}$$

for the unknown matrix $\underline{X}$.

By setting

$$\underline{Q} = \underline{I}$$

it was used to find the inverse of a matrix $\underline{A}$. The full subroutine is described in the Computation Center bulletin number CC-174-6.

CHAPTER V

COMPUTER RESULTS

A. INTRODUCTION

A number of computer runs were made to test the method and to try it out on various examples. Some of the more enlightening ones are enumerated in this chapter, together with their purposes and chief results. General discussion of results is reserved for the next chapter.

The runs are listed by plant (or state) matrix. In each case the Jordan canonical form (see e.g., Zadeh and Desoer[98] or Athans and Falb[4]) was used, with the added requirement that all the entries be real numbers.* This means normal coordinates have been used for clarity, so that the plant matrix shows the eigenvalues directly.

In this chapter an iteration of Newton's method will be called just an iteration. A step from one member of the sequence of approximate operators $\{T_k\}$ to the next will be referred to as a step. In plotting the sequence of solution vectors $\{\underline{\pi}_k\}$ the step number is indicated on the graph. Thus in Fig. 5.1 the 0 refers to the vector $\underline{\pi}_0$, the solution of the linear operator equation $T_0(\pi) = 0$; the 1 refers to the vector $\underline{\pi}_1$, the solution of the operator equation $T_1(\underline{\pi}) = 0$; the 2 refers to the vector $\underline{\pi}_2$, the solution of the operator equation $T_2(\underline{\pi}) = 0$; etc.

A summary of the runs made is given in Table 5.1. Detailed information on the sequence of approximate operators $\{T_k(\underline{\pi})\}$ (the steps and iterations) is given in Appendix C.

Some decision constants were experimented with early in the testing and then standardized at what appeared to be reasonable values.

---

* Thus a complex eigenvalue $\lambda = -a \pm jb$ leads to the form

$$\underline{A} = \begin{bmatrix} -a & b \\ -b & -a \end{bmatrix}$$

Table 5.1

Computer Results

Run No.	Name of Plant and Order	Mode of Operation ICHO	M	Computer Time Required (Seconds)	Convergence of Newton's Method	Convergence of Outer Loop	Degree of Difficulty
1	Double	0	15	Low	Good	Good	Easy
2	Integrator	2	120	Average	Good	Good	Easy
3	2	0	20	High	Poor	Fair	Difficult
4	Single	0	41	High	Fair	Good	Difficult
5	Oscillator	0	41	Low	Good	Good	Easy
6	2	0	41	Low	Good	Good	Easy
7	Damped Single	0	41	Low	Good	Good	Easy
8	Oscillator 2	18	41	Average	Good	Good	Easy
9	Damped Double	0	70	Low	Good	Good	Easy
10	Oscillator 4	0	70	Average	Good	Fair	Average
11	Double	2	40	22.9	Good	Good	Easy
12	Oscillator	2	100	48.1	Good	Good	Easy
13	4	2	100	50.2	Fair to good	Good	Average
14			100				
15		12	40	20.6	Good	Good	Easy
16		12	40	24.4	Good	Good	Easy
17		12	100	50.2	Good	Good	Easy
18							
19		10	25	13.5	Fair	Good	Average
20		10	10		Poor		Impossible
21		12	40	22.2	Fair	Good	Average to Difficult
22		12	40	41.5	Foor	Good	Difficult
23	Double Exponential	18	100	37.8	Fair	Good	Difficult
24	2	18	41	15.1	Fair	Good	Average
25		18	41	19.4	Fair	Good	Average to Difficult
26	Quadrupole	12	100	60.2	Fair to good	Good	Average
27	4	12	100	53.5	Fair to good	Good	Average
28		12	100	48.5	Fair to good	Good	Average
29		12	100	54.5	Fair to good	Good	Average
30	Quadrupole	12 10	100	58.1 44.2	Good	Good	Average
31	Oscillator 4	12	40	99.4	Good	Good	Difficult
32	Triple	12	100	109.5	Good	Good	Easy
33	Oscillator 6	10	40	40.4	Good	Good	Easy

In terms of the computer program writeup of Chapter IV, these are:

$$\begin{aligned} \text{EPMTX} &= 10^{-6} \\ \text{EPS} &= 10^{-3} \\ \text{KPETA} &= 4 \\ \text{ALPT} &= 1.0 \\ \text{AMAX} &= 10. \end{aligned}$$

The value of ICHO depends on the mode of operation desired, according to Fig. 4.3. There is a tradeoff between accuracy and computer time, but the safe method is to use a large value for the constant M, thus insuring high accuracy ($M \leq 100$).

B. DOUBLE INTEGRATOR PLANT

Two integrators in series form a plant like that of an inertial mass. With control acting on the acceleration, there results,

$$\underline{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \qquad \underline{b} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{5.1}$$

This is a very easy system to analyze. It was chosen for the first set of runs partly because the results can easily be compared with known analytic results.

Run 1

$$\underline{\xi} = \begin{bmatrix} 10 \\ 2 \end{bmatrix} \quad T = 15;\ \underline{A}, \underline{b} \text{ of Eq. } 5.1$$

Purpose: To check whether the sequence of operators chosen has the property of sequential convergence. To compare the sequence of solution vectors $\{\underline{\pi}_k\}$ with the solution vector $\underline{\pi}^*$ of the exact operator. Finally, to determine the effect of allowing η to become very large.

Results: See Fig. 5.1. The sequence of solution vectors $\{\underline{\pi}_k\}$ appear to lie on a straight line through the origin, and they also converge within the numerical accuracy used to the optimal solution

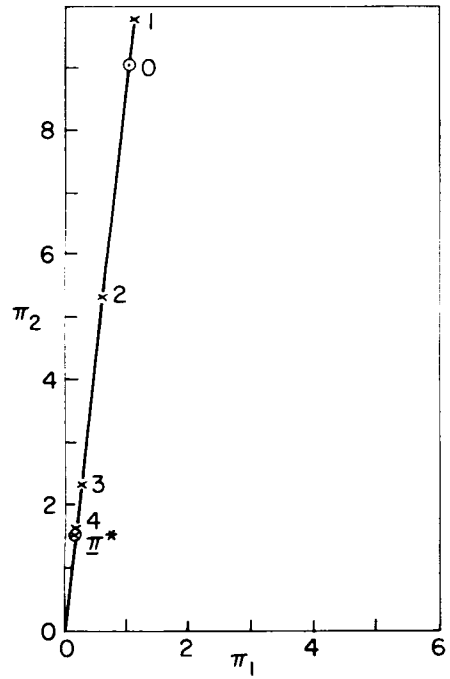


Fig. 5.1 Graph of the Sequence $\left\{\underline{\pi}_k\right\}$ - Run 1

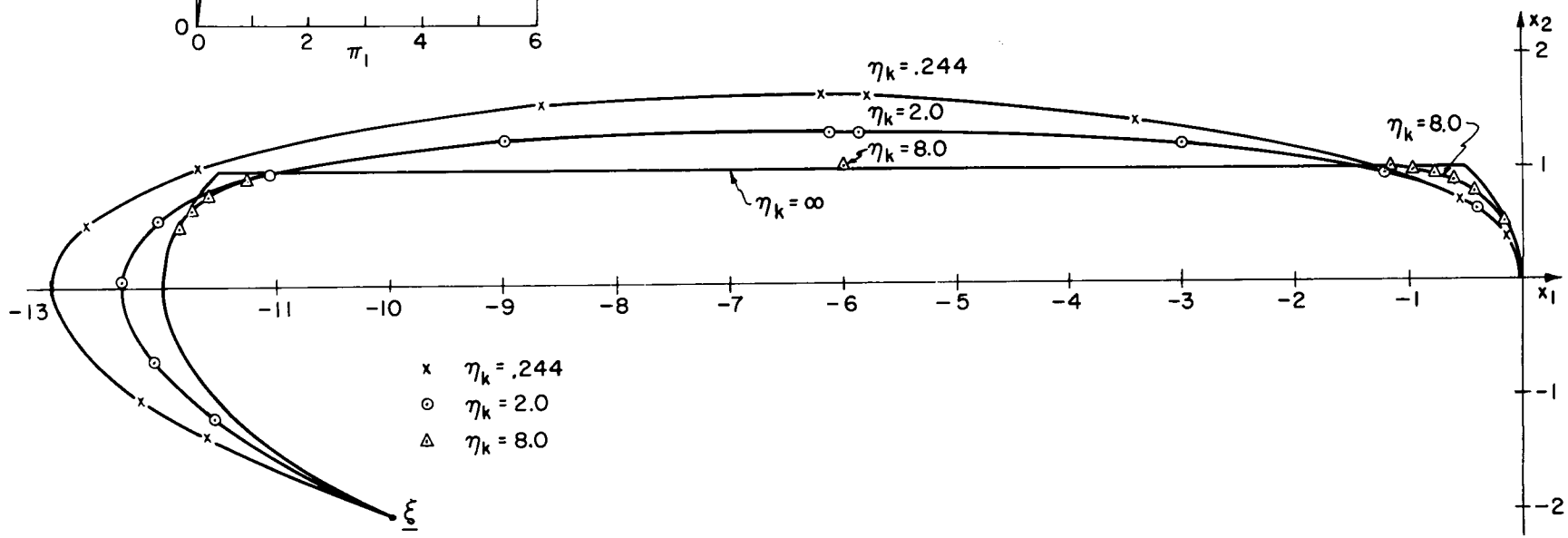


Fig. 5.2 State Space Trajectories - Run 2

vector $\underline{\pi}^*$. Sequential convergence took place at each step until η reached a value of 5,627, at which point the first derivative became too difficult to evaluate (too close to delta functions). After the sixth step (η=8.8) only one iteration of Newton's method was needed per step. This indicates that the sequence can be carried far beyond the point at which $\underline{\pi}_k$ converges to $\underline{\pi}^*$.

Run 2

$$\underline{\xi} = \begin{bmatrix} -10 \\ \\ -2 \end{bmatrix} \quad T = 15; \quad \underline{A} \text{ and } \underline{b} \text{ of Eq. 5.1}$$

Purpose: To check a symmetric initial condition to Run 1 for symmetry of results. To examine the trajectories in the state space--to compare those generated using the approximate operators $T_k(\underline{\pi})$ and the exact operator $T(\underline{\pi})$.

Results: The results seem exactly symmetrical to those in Fig. 5.1, to within a very small roundoff error. A few chosen state space trajectories are plotted in Fig. 5.2. It is apparent that as η_k increases, the trajectories approach the exact one. This plant is very revealing because it produces corners in the exact state space trajectory which are difficult to reproduce using a smooth control function. From Fig. 5.2 one can also conclude that the fuel used by the approximate controls converges very closely to the optimal value.

Run 3

$$\underline{\xi} = \begin{bmatrix} 0 \\ \\ -6 \end{bmatrix} \quad T = 15; \quad \underline{A}, \ \underline{b} \text{ of Eq. 5.1}$$

Purpose: To try a different initial condition in the state space. To examine again the upper limit for η, the property of sequential convergence, and the number of iterations per step.

Results: This run proved much more difficult than the two previous ones. The minimum possible time to reach the origin from this initial condition is $T^* = 6(1+\sqrt{2}) = 14.48$, so this run is very close to the minimum time solution. In Run 1, the minimum possible time

was only $T^* = 8.93$. At several steps the program required two attempts to define the next operator $T_k(\underline{\pi})$.

Just as in Run 1, the vector $\underline{\pi}_k$ was very close to its final value by the step where η_k was equal to 5 or 10. In the steps after that very few iterations were needed. However in this difficult problem it still sometimes required two attempts to define the next operator $T_k(\underline{\pi})$. Numerical difficulties seem to have set in when η exceeds 19,450; probably due to the very inaccurate first derivative operator obtained, Newton's method suddenly diverges.

C. SINGLE OSCILLATOR PLANT

A single degree of freedom oscillator without damping has the system matrix below.

$$\underline{A} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad \underline{b} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{5.2}$$

This plant was chosen first of all because many physical problems can be modelled by the spring and mass system. Secondly, because it prepares for later work with a two degree of freedom oscillator. Finally, this leads to variety; first a plant was used with poles at the origin and now one with poles on the imaginary axis.

Runs 4, 5, and 6 were terminated as soon as $\eta_k > 2$.

Run 4

$$\underline{\xi} = \begin{bmatrix} -2 \\ \\ 2 \end{bmatrix} T = 4.7124 = \frac{3\pi}{2} \; ; \; \underline{A}, \underline{b} \text{ of Eq. 5.2}$$

Purpose: To try out a different plant.

Results: The minimum time solution requires a time of $T^* = 2\pi - 2\text{Tan}^{-1} 2 \approx \frac{3\pi}{2} - .643$, so this is also a difficult problem. At most of the steps, two attempts were needed to define the next operator $T_k(\underline{\pi})$.

Run 5

$$\underline{\xi} = \begin{bmatrix} 3. \\ \\ 3. \end{bmatrix} \quad T = 9.4248 = 3\pi; \ \underline{A}, \underline{b} \text{ of Eq. 5.2}$$

Purpose: To try another initial conditon.

Results: In this run the problem is not as difficult as Run 4, $T^* = 3\pi - \frac{\pi}{2} - \text{Sin}^{-1} \frac{1}{\sqrt{13}} - \text{Tan}^{-1} \frac{3}{2} \approx 3\pi - 2.835$. Only once was it necessary to redefine an operator $T_k(\underline{\pi})$. The most difficult step was the second one, which required redefining the operator T_2 and then needed five iterations for convergence.

The sequence of solution vectors $\{\underline{\pi}_k\}$ plotted in Fig. 5.3 still lies on a straight line.

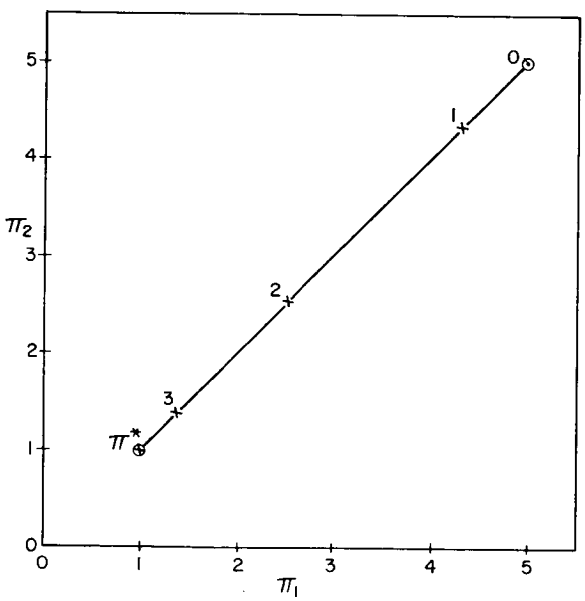


Fig. 5.3 Graph of Sequence $\{\underline{\pi}_k\}$ - Run 5

Run 6

$$\underline{\xi} = \begin{bmatrix} 0. \\ \\ 2. \end{bmatrix}, \quad T = 6.0; \ \underline{A}, \underline{b} \text{ of Eq. 5.2}$$

Purpose: To try another initial condition.

Results: This run was easy. No redefining of operators was needed.

D. DAMPED OSCILLATOR PLANT

In this section tests are described on four different plants. Two are single oscillators and two are double oscillators, all with real (negative) damping.

The overall purpose is to try some tests on plants whose roots have negative real parts.

A secondary purpose is to try a higher order plant. For plotting purposes the vector $\underline{\pi}_k$ is split into two vectors of two elements each.

$$\underline{\pi}_{12} = \begin{bmatrix} \pi_1 \\ \\ \pi_2 \end{bmatrix}$$

$$\underline{\pi}_{34} = \begin{bmatrix} \pi_3 \\ \\ \pi_4 \end{bmatrix}$$

Then π_2 is plotted against π_1, and as a separate graph π_4 is plotted against π_3.

Run 7*

$$\underline{A} = \begin{bmatrix} -.1 & 1 \\ \\ -1 & -.1 \end{bmatrix}, \quad \underline{b} = \begin{bmatrix} 0 \\ \\ 1 \end{bmatrix}$$

$$\underline{\xi} = \begin{bmatrix} -2 \\ \\ 2 \end{bmatrix} \quad T = 4.7124 = \frac{3}{2}\pi$$

---

* The usual differential equation for this plant is

$$\ddot{y}(t) + .2\,\dot{y}(t) + 1.01\,y(t) = u(t)$$

Purpose: To test the method with damping present.

Results: One redefining of an operator was required. The total number of iterations needed was 21.

The vectors $\underline{\pi}_k$ no longer lie on a straight line; they are slightly off.

Run 8

$$\underline{A} = \begin{bmatrix} -.1101 & 1 \\ -1 & -.1101 \end{bmatrix} \quad \underline{b} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\underline{\xi} = \begin{bmatrix} .5 \\ .5 \end{bmatrix} \quad T = 6.2832 = 2\pi$$

Purpose: To try another initial condition. To test the subroutine CKCON. To compare the fuel costs.

Results: The trajectories in the state space were found to be too close together to be worth plotting for comparison.

As η was increased the fuel used ranged from 1.88 units down to 1.46 units, a 23 percent decrease.

The parameter h from the convergence theorem of Kantorovich was too high to guarantee convergence (convergence is guaranteed for $h < 1/2$), even though convergence did occur at each step. The estimate of convergence h_1 had lower values, but was also generally larger than $\frac{1}{2}$. As shown in Fig. 5.5, there appears to be some connection between the value of h_1 (or h) for an operator $T_{k+1}(\underline{\pi})$ and the number of iterations of Newton's method required for convergence to the solution of that operator. See also the results for the Double Exponential Plant.

The sequence of vectors $\{\underline{\pi}_k\}$ appears to lie nearly in a straight line in Fig. 5.4.

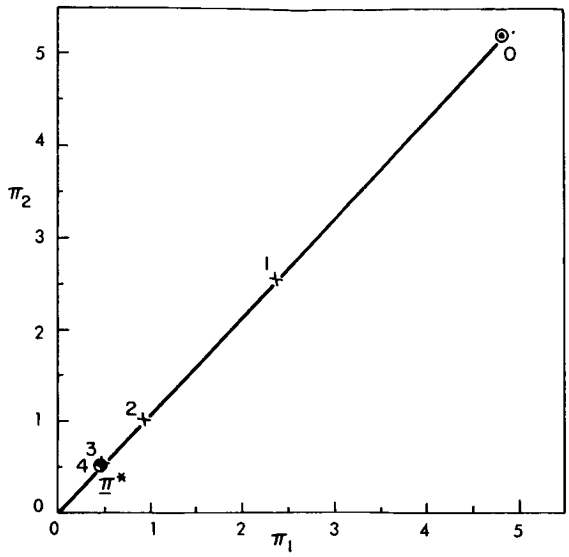


Fig. 5.4 Graph of Sequence $\{\underline{\pi}_k\}$ - Run 8

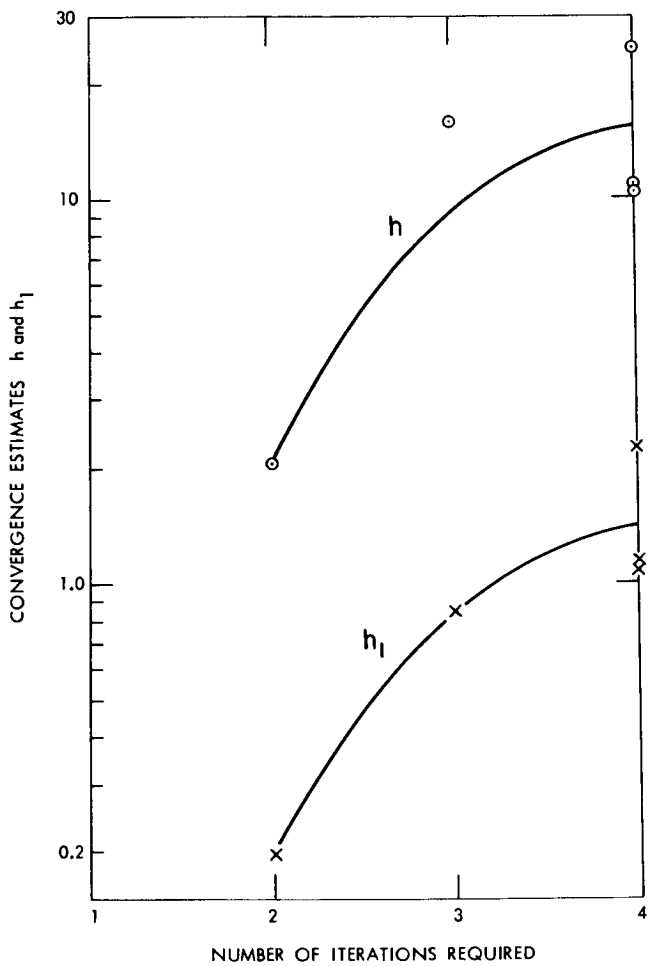


Fig. 5.5 Convergence Parameters vs. Number of Iterations - Run 8

Run 9

$$\underline{A} = \begin{bmatrix} -.1101 & 1 & & \\ -1 & -.1101 & & 0 \\ & & -.1101 & 2 \\ 0 & & -2 & -.1101 \end{bmatrix} \quad \underline{b} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

$$\underline{\xi} = \begin{bmatrix} 1. \\ 1. \\ 3. \\ 3. \end{bmatrix} \quad T = 6.2832 = 2\pi$$

Purpose: To try out the program on a plant with a four dimensional state space.

Results: See Fig. 5.6. The vectors $\underline{\pi}_k$ definitely do not lie on straight lines.

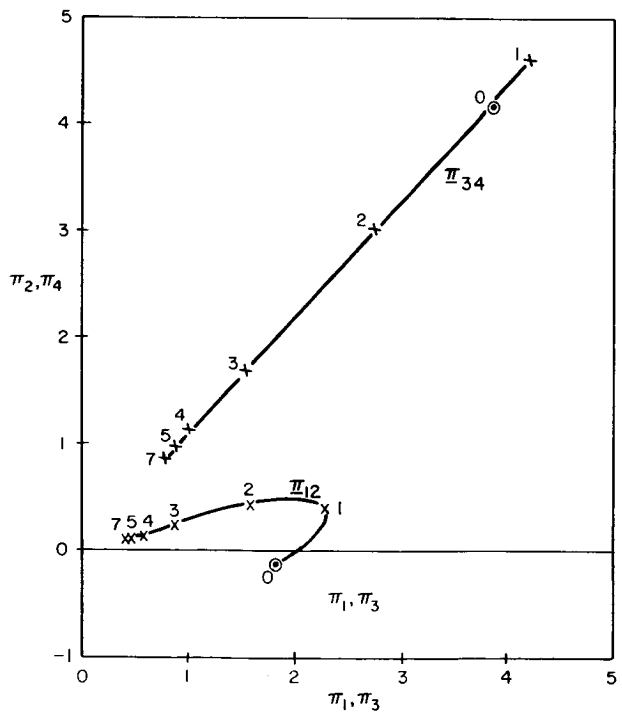


Fig. 5.6 Graph of the Sequence $\{\underline{\pi}_k\}$ - Run 9

None of the operators had to be redefined.

Run 10

$$\underline{A} = \begin{bmatrix} -.5 & 5. & & \\ -5. & -.5 & 0 & \\ & & -.6 & 10. \\ 0 & & -10. & -.6 \end{bmatrix} \quad \underline{b} = \begin{bmatrix} 0 \\ .5 \\ 0 \\ .6 \end{bmatrix}$$

$$\underline{\xi} = \begin{bmatrix} 10. \\ 10. \\ 10. \\ 10. \end{bmatrix} \quad T = 8$$

Purpose: To try a problem for which the state of the system has many oscillations.

Results: The program in this case was somewhat conservative in choosing the sequence $\{\eta_k\}$, so that no more than four iterations were needed at any one step. By the time $\eta_k \approx 1.$, the vector $\underline{\pi}_k$ had become very close to its final value.

E. DOUBLE OSCILLATOR PLANT

This plant is characterized by the matrix

$$\underline{A} = \begin{bmatrix} 0 & 1 & & \\ -1 & 0 & 0 & \\ & & 0 & \omega \\ 0 & & -\omega & 0 \end{bmatrix} \quad \underline{b} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} \tag{5.3}$$

It can be described either as two single degree of freedom oscillators having a common control or as a single oscillator with two degrees of freedom. Note that the frequency ω is left as a parameter.

At this point the basic features of the program have all been tested. Now questions of accuracy and some features of the problems themselves will be examined.

There are four series of runs using this plant. Each series will be described separately. A vector $\underline{\ell}$ is defined for use in these runs.

$$\underline{\ell} = \begin{bmatrix} \sqrt{3} \\ 1 \\ 1 \\ \sqrt{3} \end{bmatrix}$$

1. Effect of Varying ω - Runs 11-14

In this series the frequency ω is varied. As $\omega \rightarrow 1$, the two oscillators become increasingly alike and therefore more difficult to handle with one control.

$$\underline{\xi} = 2\underline{\ell}, \quad T = 12.5664 = 4\pi \tag{5.4}$$

The purpose is to examine the changes in the sequence of approximate operators as the problem becomes more difficult.

A plot of the optimal control variable time history was also made to illustrate how the nature of the control changes as the problem becomes more difficult.

Run 11: $\omega = 4$, $M = 40$; Eqs. 5.3 and 5.4

Figure 5.7 shows the fuel optimal control history. The rapid switching pattern is taken from the fuel optimal control for one oscillator with $\omega = 4$. This is superimposed on a slower pattern, representing the control for one oscillator with $\omega = 1$.

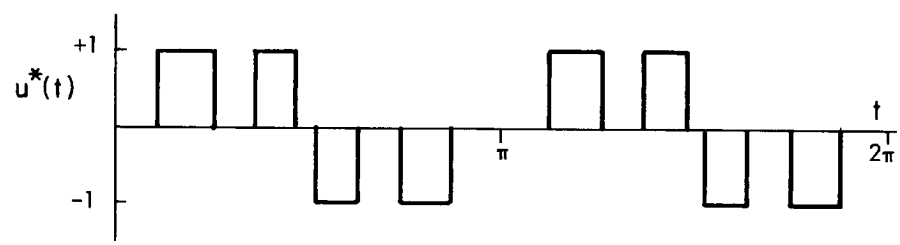


Fig. 5.7 Graph of Fuel Optimal Control vs. Time Run 11

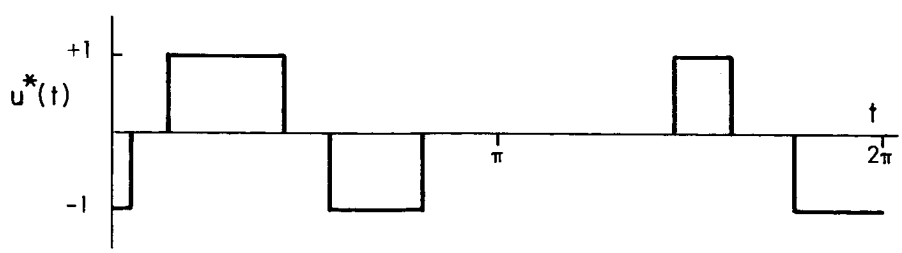


Fig. 5.8 Graph of Fuel Optimal Control vs. Time Run 12

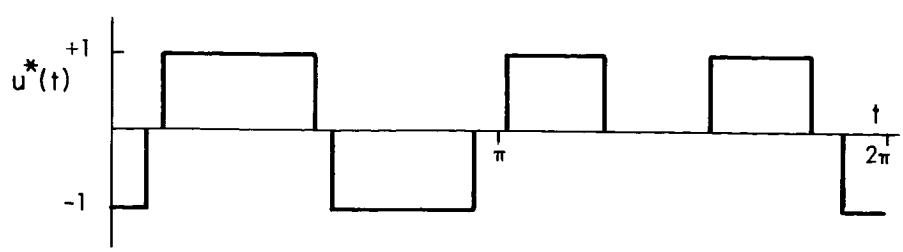


Fig. 5.9 Graph of Fuel Optimal Control vs. Time Run 13

Run 12: $\omega = 1.5$, $M = 100$; Eqs. 5.3 and 5.4

In Fig. 5.8 the fuel optimal control history still has somewhat the same pattern as in Fig. 5.7 but it is obscured by the closeness of the frequencies.

Run 13: $\omega = 1.3$, $M = 100$; Eqs. 5.3 and 5.4

In Fig. 5.9 it is no longer possible to detect the characteristic pattern of Fig. 5.7. Notice that the control is on most of the time.

Run 14: $\omega = 1.2$, $M = 100$

The program was unable to reduce the slope a to zero, making this either a very difficult problem or else an impossible one ($\underline{\xi}$ does not belong to the set of reachable states).

In Runs 11 and 12 the total fuel used is very nearly the same. As ω is further decreased to 1.3 in Run 13 the total fuel used rises sharply, and for $\omega = 1.2$ the problem does not seem to have a solution. This shows how the problem difficulty suddenly rises as ω gets close to one.

2. Effect of Varying $\|\underline{\pi}^*\|$ Runs 15-18

These runs were made to explore the relationship between the initial state vector $\underline{\xi}$ and the optimal costate initial condition vector $\underline{\pi}^*$. In addition the routine for computing $\underline{\xi}$ when given $\underline{\pi}^*$ was checked.

$$\omega = 4., \quad T = 4\pi \tag{5.5}$$

Run 15: $\underline{\pi}^* = .5\underline{\ell}$, $M = 40$; Eqs. 5.3 and 5.5

The sequence of vectors $\{\underline{\pi}_k\}$ converges to the true vector $\underline{\pi}^*$ to within the numerical accuracy used as shown in Fig. 5.10. The length of the computed vector $\underline{\xi}$ is,

$$\|\underline{\xi}\|_2 = 5.16$$

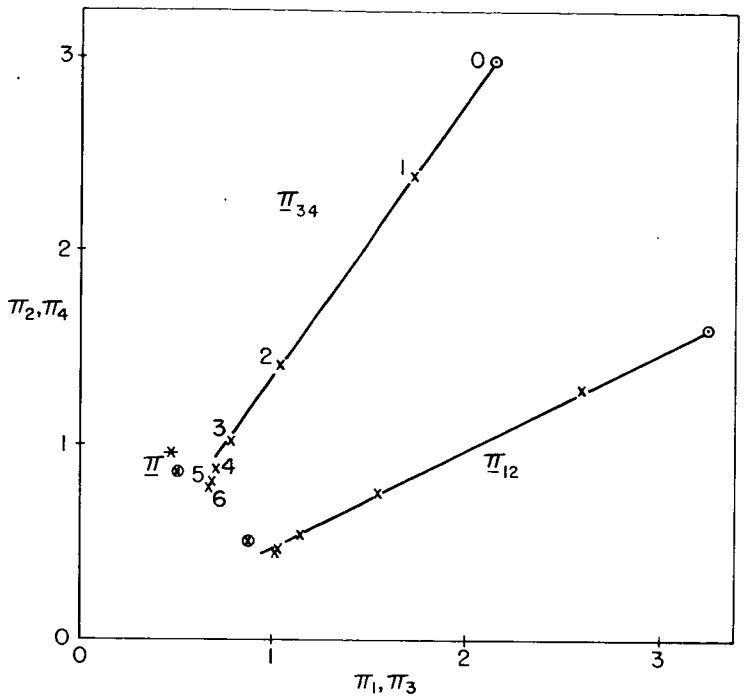


Fig. 5.10 Graph of the Sequence $\{\underline{\pi}_k\}$ - Run 15

Run 16: $\underline{\pi}^* = \underline{\ell}$, M = 40; Eqs. 5.3 and 5.5

Again the sequence of vectors $\{\underline{\pi}_k\}$ converges to an area around the true vector $\underline{\pi}^*$.

The length of the computed vector $\underline{\xi}$ is,

$$\|\xi\|_2 = 6.50$$

Run 17: $\underline{\pi}^* = 2\underline{\ell}$, M = 100; Eqs. 5.3 and 5.5

This time the sequence of vectors $\{\underline{\pi}_k\}$ converges more closely to the true vector $\underline{\pi}^*$, due to the greater numerical accuracy used.

$$\|\underline{\xi}\|_2 = 6.91$$

Run 18: $\underline{\pi}^* = 4\underline{\ell}$, M = 100; Eqs. 5.3 and 5.5

$$\|\underline{\xi}\|_2 = 7.21$$

Conclusions: All of these runs are in the easy category. The slope α was reduced to zero in one step and none of the operators had to be redefined.

A plot of the initial condition vectors $\underline{\xi}$ is shown in Fig. 5.11. It is apparent that the repeated doubling of $\|\underline{\pi}^*\|$ leads to diminishing increases in $\|\underline{\xi}\|$ as the available control effort becomes used up.

A plot of the fuel used versus $\log_{10}\eta$ for these runs is shown in Fig. 5.12. Notice how well the curves converge to the optimum values.

3. Effect of Decreased Accuracy - Runs 19-20

A check was made of the effect of decreased accuracy on the procedure. This is done by decreasing M, which makes the integration step size larger.

$$\omega = 1.5,\ T = 2\pi,\ \underline{\pi}^* = 2\underline{\ell} \tag{5.6}$$

Run 19: M = 25; Eqs. 5.3 and 5.6

A total of 33 iterations of Newton's method were required, more than was needed for most of the higher accuracy runs using this plant. Computer time consumed was 13.7 seconds. The sequence of vectors $\{\underline{\pi}_k\}$ in Fig. 5.13 should be compared with the ones in Fig. 5.10 or perhaps Fig. 5.4 or 5.21. The sequence converges most closely to $\underline{\pi}^*$ when M = 100, and rather far from $\underline{\pi}^*$ in this run.

$$\underline{\xi} = \begin{bmatrix} 4.4 \\ 2.50 \\ 2.22 \\ 4.27 \end{bmatrix}$$

Run 20: M = 10; Eqs. 5.3 and 5.6

In this case the accuracy is so bad that the program quits. The first operator does not lead to convergence in Newton's method, and the vector $\underline{\pi}$ soon becomes so large that all eleven mesh points have control of nearly 1.0 in magnitude. Then the first derivative no longer has an inverse and the program quits.

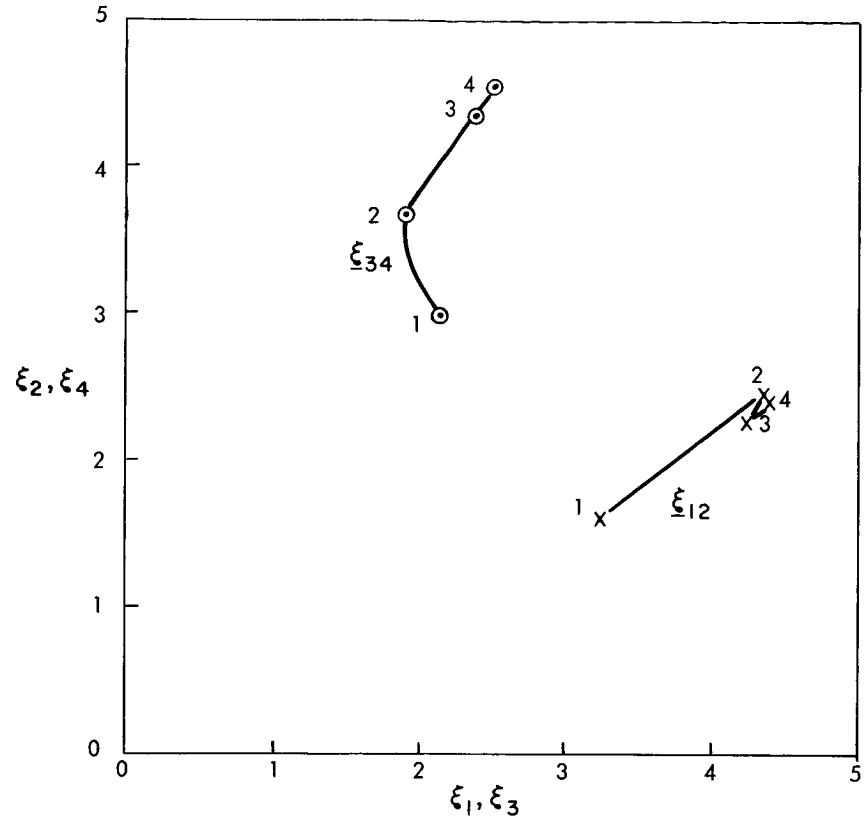


Fig. 5.11 Computed Initial Condition Vectors $\underline{\xi}$ Runs 15-18

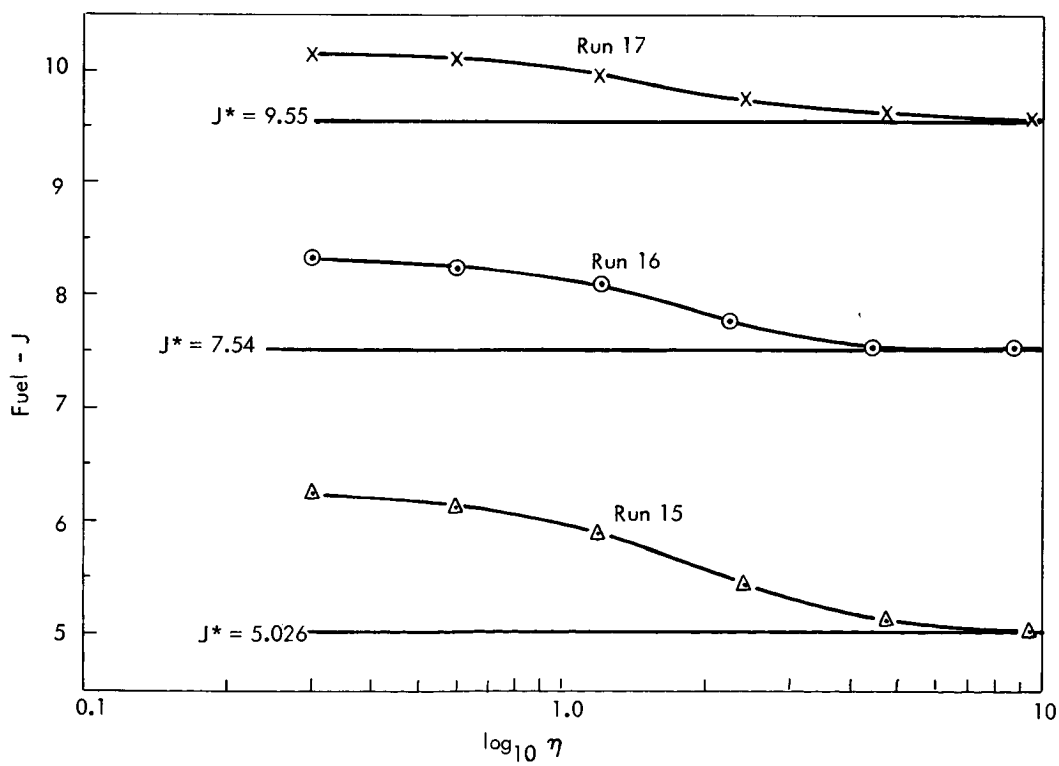


Fig. 5.12 Graphs of Total Fuel Used vs log. η_k for the Double Oscillator Plant - $\underline{\pi}$ * Varied Runs 15-17

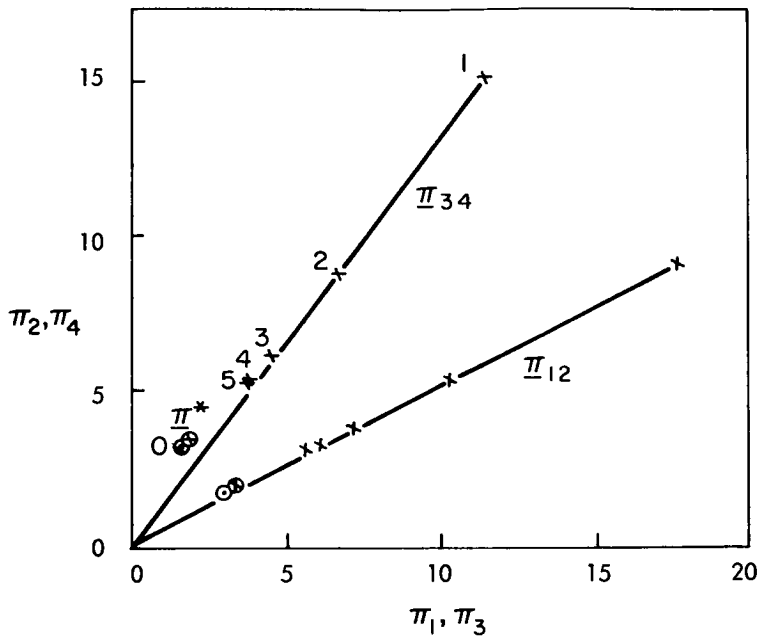


Fig. 5.13 Graph of the Sequence $\{\underline{\pi}_k\}$ Run 19

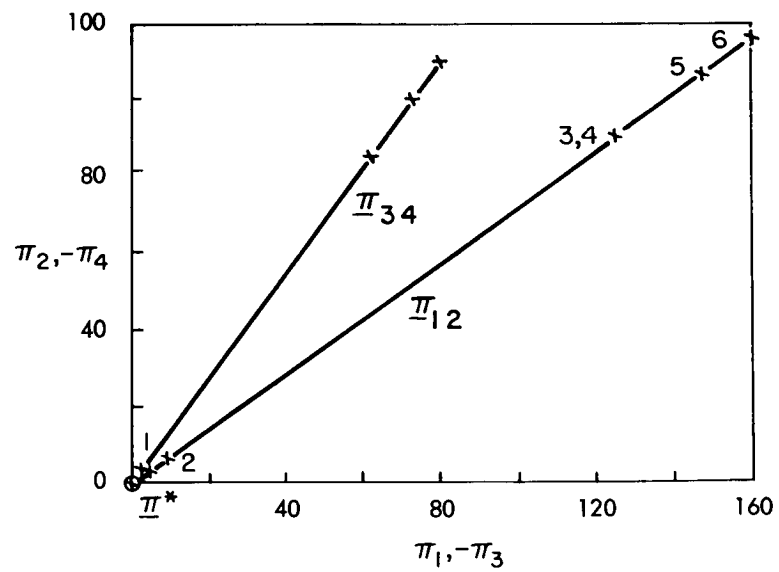


Fig. 5.14 Graph of the Sequence $\{\underline{\pi}_k\}$ Run 22

The iterations show a drifting pattern characteristic of some very difficult runs. That is, instead of converging or diverging quadratically, the vector $\underline{\pi}_i$ changes by almost a constant amount at each iteration.

$$\underline{\xi} = \begin{bmatrix} 5.06 \\ 2.42 \\ 2.67 \\ 4.93 \end{bmatrix}$$

4. Effect of Nonunique $\underline{\pi}^*$ - Runs 21-22

As noted in Chapter III, Lemma 3.1, if the vectors $\underline{q}(t_i)$, $i = 1, 2, \ldots m$, do not span the space R_n, then the costate initial condition vector $\underline{\pi}^*$ is not uniquely specified. With this plant, $T > \pi$ insures uniqueness, while $T \leq \pi/2$ leads to less than n switchings, and hence to a nonunique $\underline{\pi}^*$. More precisely, two switchings always occurred, leaving two degrees of freedom open. Another test of this type was carried out on the quadrupole plant.

$$T = \pi/2 \ , \quad M = 40, \ \underline{\pi}^* = .5\underline{\ell} \tag{5.7}$$

Run 21: $\omega = 2$; Eqs. 5.3 and 5.7

The sequence of vectors $\{\underline{\pi}_k\}$ still lies on a straight line, but instead of converging inward toward the given $\underline{\pi}^*$ the sequence moves far out. Newton's method seems to have had no special difficulty in converging, and the run seems about equal in difficulty to others of its type with a unique $\underline{\pi}^*$.

Run 22: $\omega = 1.5$; Eqs. 5.3 and 5.7

The sequence of solution vectors $\{\underline{\pi}_k\}$ is shown in Fig. 5.14. The linear term was $a_0 = -1.31$ and three operators were needed to remove it. The median value of a_0 for runs using this plant was between -.20 and -.21. Generally this run was a bit more difficult, but similar to Run 21. The graph was made for this run, since it is the more extreme case.

Figure 5.15 shows that the sequence of approximate controls still converges, even though $\| \underline{\pi}_k \|_2$ is increasing without any apparent bound. The conclusion is that the method seems to work all right with nonunique $\underline{\pi}^*$.

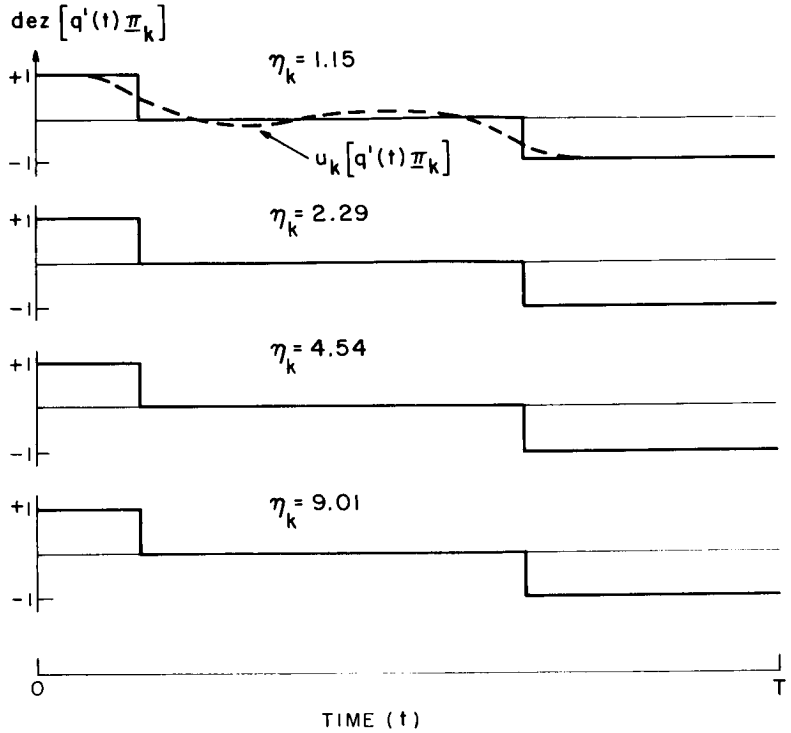


Fig. 5.15 Graph of Fuel Optimal Control vs. Time - Run 22

F. DOUBLE EXPONENTIAL PLANT

This plant has two real poles, one of them unstable.

$$A = \begin{bmatrix} 1. & 0 \\ 0 & -1. \end{bmatrix} \qquad B = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \qquad T = 2.0 \tag{5.8}$$

The main purpose is to try out the procedure with an unstable pole to see if any numerical difficulties are introduced. A second purpose is to try out the procedure with poles on the real axis.

This plant has been analyzed in terms of the initial costate vector $\underline{\pi}^*$, as shown in Fig. 5.16. The sequence of controls shown may not

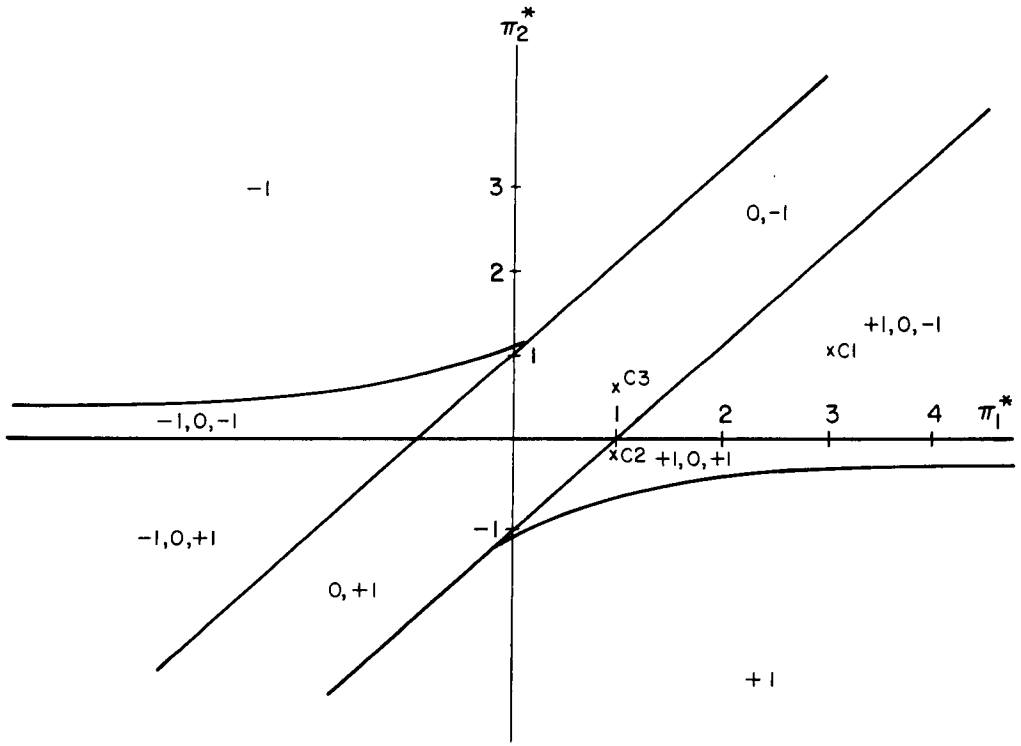


Fig. 5.16 Graph of Control Sequence Regions of $\underline{\pi}^*$ for the Double Exponential Plant

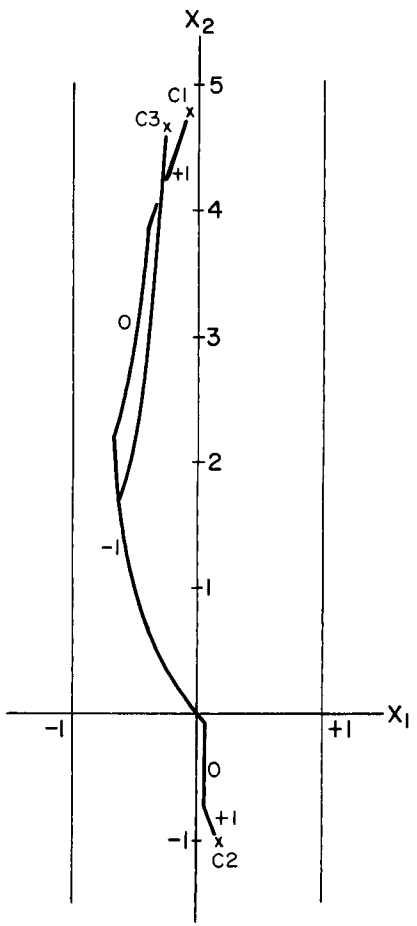


Fig. 5.17 State Space Trajectories for Fuel-Optimal Control of Double Exponential Plant

be completed if the time T is too small. Thus if

$$\underline{\pi}^* = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

the indicated sequence of control history is +1, 0, -1. However for $T < \ell n\ .5(\sqrt{13} - 1)$ only the control +1 occurs.

In Fig. 5.17 a couple of typical trajectories in the state space are shown. The state cannot be brought to the origin unless $|\xi_1| < 1$. In practice the optimal costate initial condition $\underline{\pi}^*$ was specified, to insure that the vector $\underline{\xi}$ would belong to the set of states reachable at time T. As mentioned in Chapter IV, the vector $\underline{\pi}^*$ is then discarded.

A third purpose was to try out the subroutine CKCON again and to examine the convergence criteria. Figure 5.18 and Fig. 5.19 give these results. See also Run 8.

Figure 5.20 is a plot of the fuel used versus $\log_{10}\eta$ for these runs, which shows that the suboptimal controls come close to the optimal in conserving fuel.

Run 23: $\underline{\pi}^* = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$ M = 100; see Eq. 5.8

This costate initial condition was chosen to give a control history of +1, 0, -1. The sequence of vectors $\{\underline{\pi}_k\}$ in Fig. 5.21 converged nicely to the true value $\underline{\pi}^*$. Notice that they do not quite lie in a straight line. They seem to first increase on one line until $a \to 0$ and then decrease on a line through the origin. So the linear solution is not proportional to $\underline{\pi}^*$.

This run was rather difficult numerically, in spite of the high accuracy used. Some indication of this is seen in the large magnitudes of the intermediate vectors of the sequence $\{\underline{\pi}_k\}$. Two operators had to be redefined. The 37.8 seconds of computer time used was a large amount for a two-state problem.

In Fig. 5.19 the estimated convergence parameter h_1 is plotted for each operator as a function of the number of iterations of Newton's method required to converge to its solution. The largest values of h_1

Run No.	η	h_1	h	Iterations
23	$.614^{(-1)}$	$.782^{(-1)}$	$.498^{(1)}$	2
	.163	$.123^{(2)}$	$.196^{(4)}$	5
	.298	$.137^{(3)}$	$.454^{(6)}$	8
	.544	$.985^{(2)}$	$.271^{(6)}$	6
	$.144^{(1)}$	$.137^{(3)}$	$.524^{(6)}$	6
	$.384^{(1)}$	$.359^{(2)}$	$.142^{(6)}$	4
24	.106	$.288^{(2)}$	$.615^{(3)}$	4
	.282	$.101^{(2)}$	$.252^{(3)}$	4
	.748	.541	$.132^{(3)}$	3
	$.198^{(1)}$	$.111^{(2)}$	$.390^{(3)}$	4
	$.363^{(1)}$	$.121^{(2)}$	$.458^{(3)}$	4
	$.963^{(1)}$	$.988^{(1)}$	$.509^{(3)}$	4
25	$.365^{(-1)}$	.218	$.590^{(1)}$	2
	$.97^{(-1)}$	$.321^{(1)}$	$.281^{(3)}$	3
	.177	$.218^{(2)}$	$.832^{(4)}$	6
	.324	$.193^{(2)}$	$.674^{(4)}$	6
	.86	$.268^{(2)}$	$.158^{(5)}$	6
	$.228^{(1)}$	$.13^{(2)}$	$.55^{(4)}$	4
	$.606^{(1)}$	$.727^{(2)}$	$.157^{(6)}$	5

Fig. 5.18 Values of Convergence Parameters h_1 and h
Double Exponential Plant

Note: Superscripts indicate powers of 10
$.39^{(3)} = 390$

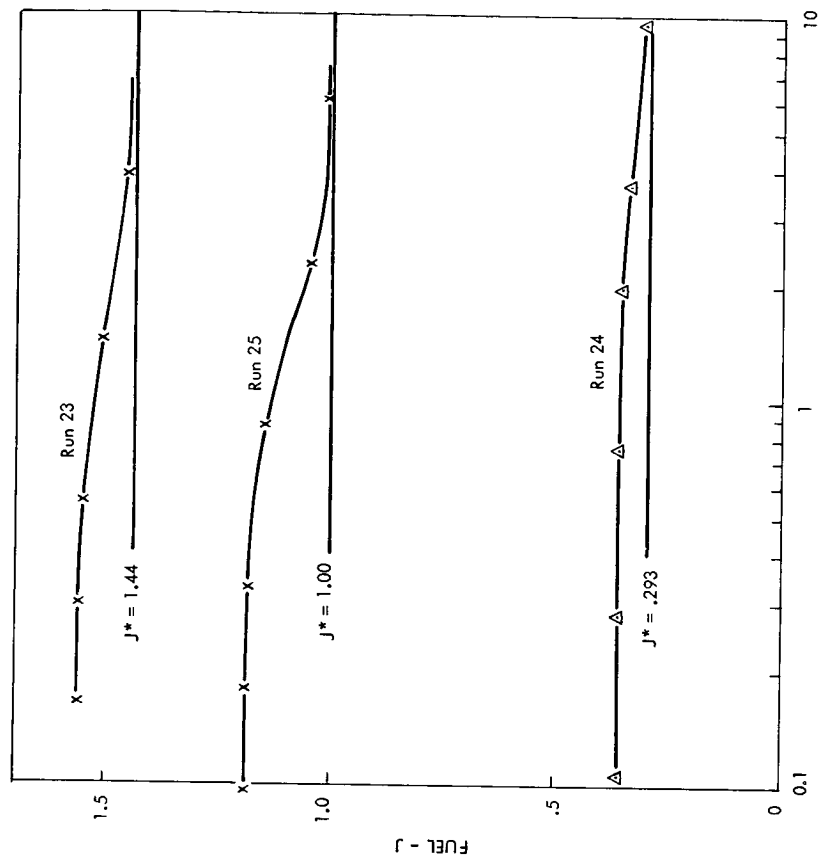


Fig. 5.20 Graph of Total Fuel Used vs. $\log_{10} \eta_k$ - Runs 23-25

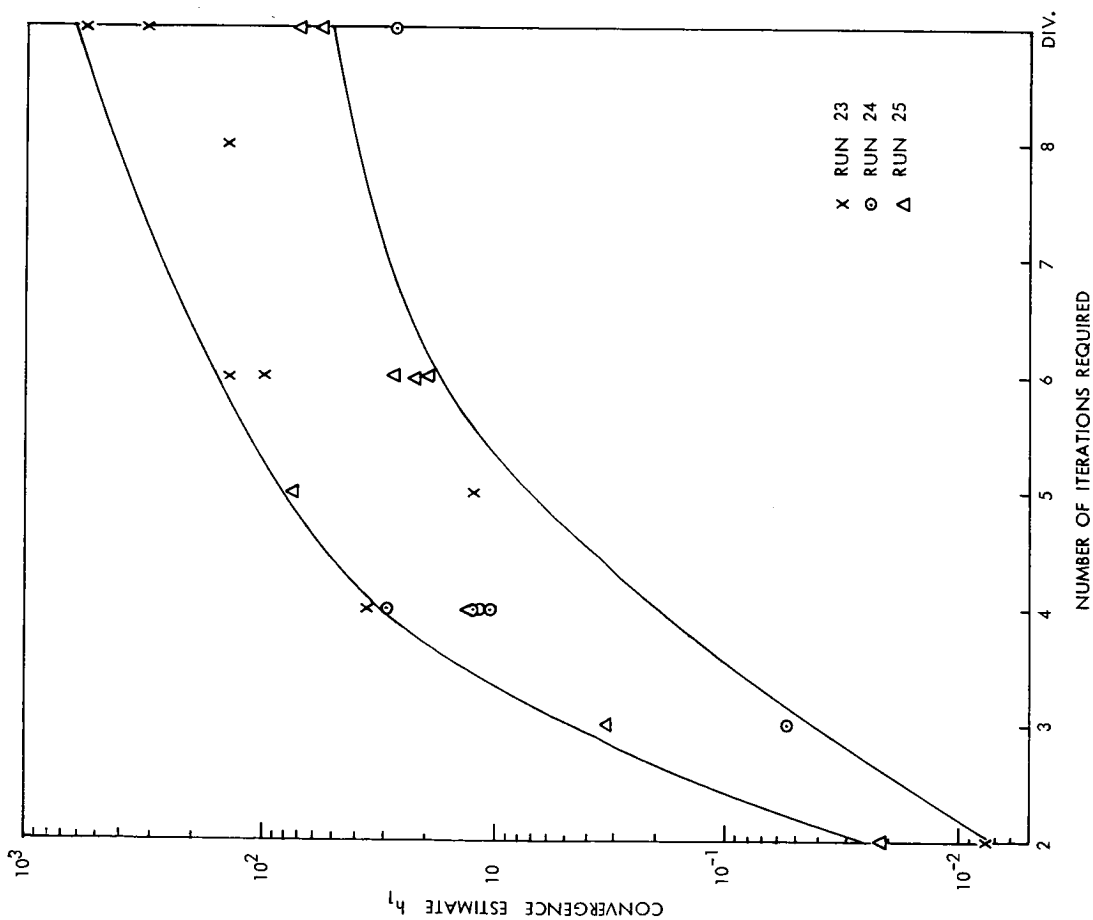


Fig. 5.19 Graph of Convergence Parameter h_1 vs. Number of Iterations - Runs 23-25

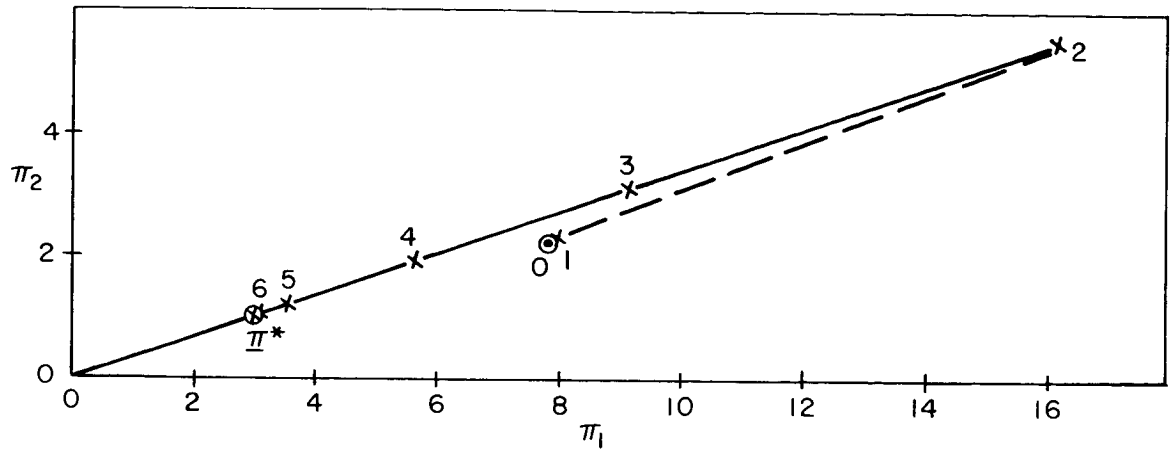


Fig. 5.21 Graph of the Sequence $\{\underline{\pi}_k\}$ Run 23

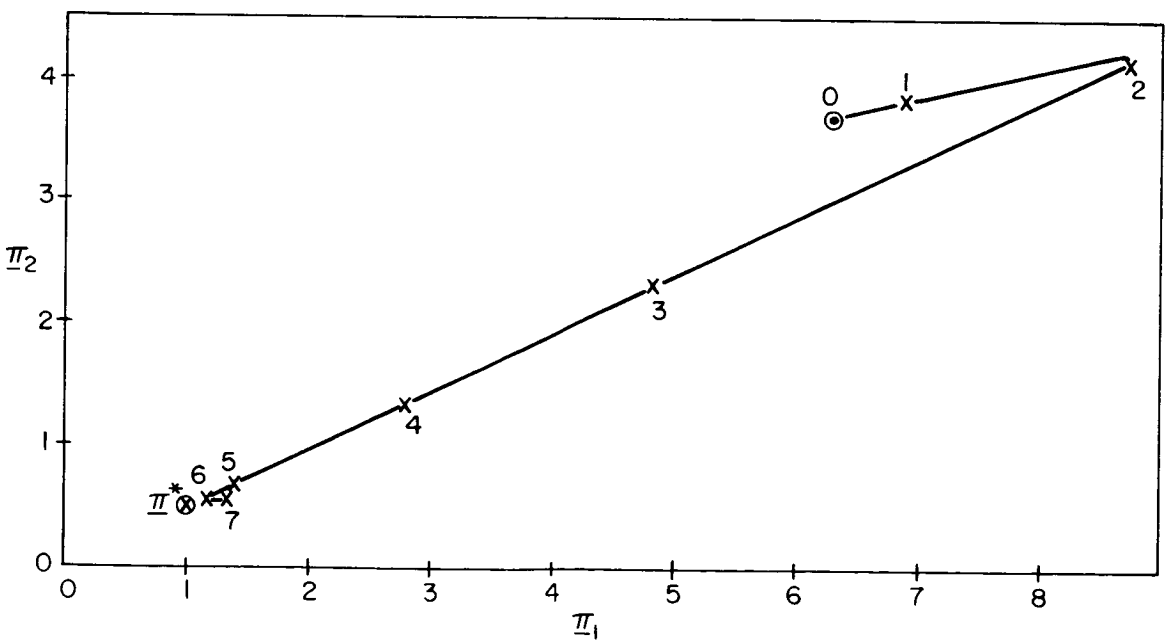


Fig. 5.22 Graph of the Sequence $\{\underline{\pi}_k\}$ Run 25

were recorded for the two operators which lead to divergence in Newton's method. There is also a tendency for low values of h_1 to correspond to faster convergence of Newton's method. However this does not always hold.

Run 24: $\underline{\pi}^* = \begin{bmatrix} 1 \\ -.13 \end{bmatrix}$ $M = 41$; see Eq. 5.8

This run was chosen to have the control sequence +1, 0, +1. It was much easier numerically than the previous one. One operator had to be redefined. The sequence of vectors $\{\underline{\pi}_k\}$ first moved in beyond the optimal value $\underline{\pi}^*$, then came back out to it.

In Fig. 5.19 the convergence parameter h_1 is generally lower for a given number of iterations in this run than in either of the others. Thus an operator with $h_1 = 136$ led to convergence in the last run, while one with $h_1 = 26$ led to divergence in this run. So the smaller values of h_1 tend to indicate easier convergence among the operators of a given run, but this property does not always apply to comparisons between runs.

Run 25: $\underline{\pi}^* = \begin{bmatrix} 1. \\ .5 \end{bmatrix}$, $M = 41$; see Eq. 5.8

This run was chosen to have the control sequence 0, -1. So the vector $\underline{q}(t_1)$ at the switch time t_1 obviously does not span the space R_2. The sequence of vectors $\{\underline{\pi}_k\}$ in Fig. 5.22 starts on one line until $\alpha \to 0$, then approaches the vector $\underline{\pi}^*$ along a line through the origin. However in this run the last two vectors $\underline{\pi}_k$ turned away from $\underline{\pi}^*$, along a third line. So possibly the nonuniqueness of $\underline{\pi}^*$ had no effect on the sequence until it came close to $\underline{\pi}^*$ and could begin moving in the correct direction from $\underline{\pi}^*$. The overall graph looks somewhat like a letter z.

In this run the largest values of h and h_1 occurred at the last operator, and these values of h and h_1 led to convergence, even though divergence occurred for two earlier operators with lower values of h and h_1. Perhaps when $\underline{\pi}_k$ is not completely constrained, it is easier for Newton's method to converge.

In difficulty this run was about equal to Run 24, again indicating that the method works all right with nonunique $\underline{\pi}^*$.

G. THE QUADRUPOLE PLANT

With a combination of the double exponential plant and the single oscillator plant, a symmetric arrangement of four poles is obtained.

$$\underline{A} = \begin{bmatrix} 1 & 0 & & 0 \\ 0 & -1 & & \\ & & 0 & 1 \\ 0 & & -1 & 0 \end{bmatrix}, \quad \underline{b} = \begin{bmatrix} 1 \\ -1 \\ 0 \\ 1 \end{bmatrix} \qquad T = 2.0 \qquad (5.9)$$

$$M = 100$$

When $\omega = 1$ in the oscillator portion this plant was called a quadrupole, having a Butterworth pattern of poles. For other values of ω the plant was called a quadrupole oscillator--the case $\omega = 4$ is treated in the next section.

One purpose of this section was to examine the effect of having too few switchings to span the space R_4.

This plant with a real pole, an unstable pole, and an oscillator is a rather general combination. Many other combinations could be made of course, but this will be the most complex example studied here. It is felt that additional poles and more complex arrangements will be increasingly difficult to analyze, and will not contribute much new information.

A couple of typical trajectories are shown in Fig. 5.23. A plot of fuel used versus η for these runs is shown in Fig. 5.24.

$$\underline{\text{Run 26}}: \ \underline{\pi}^* = \begin{bmatrix} 0 \\ -2 \\ \\ 10 \\ 0 \end{bmatrix} \ ; \ \text{see Eq. 5.9}$$

This initial costate vector was chosen to give enough switchings to span the space R_4 and make $\underline{\pi}^*$ unique. The control history is -1, 0, +1, 0, -1.

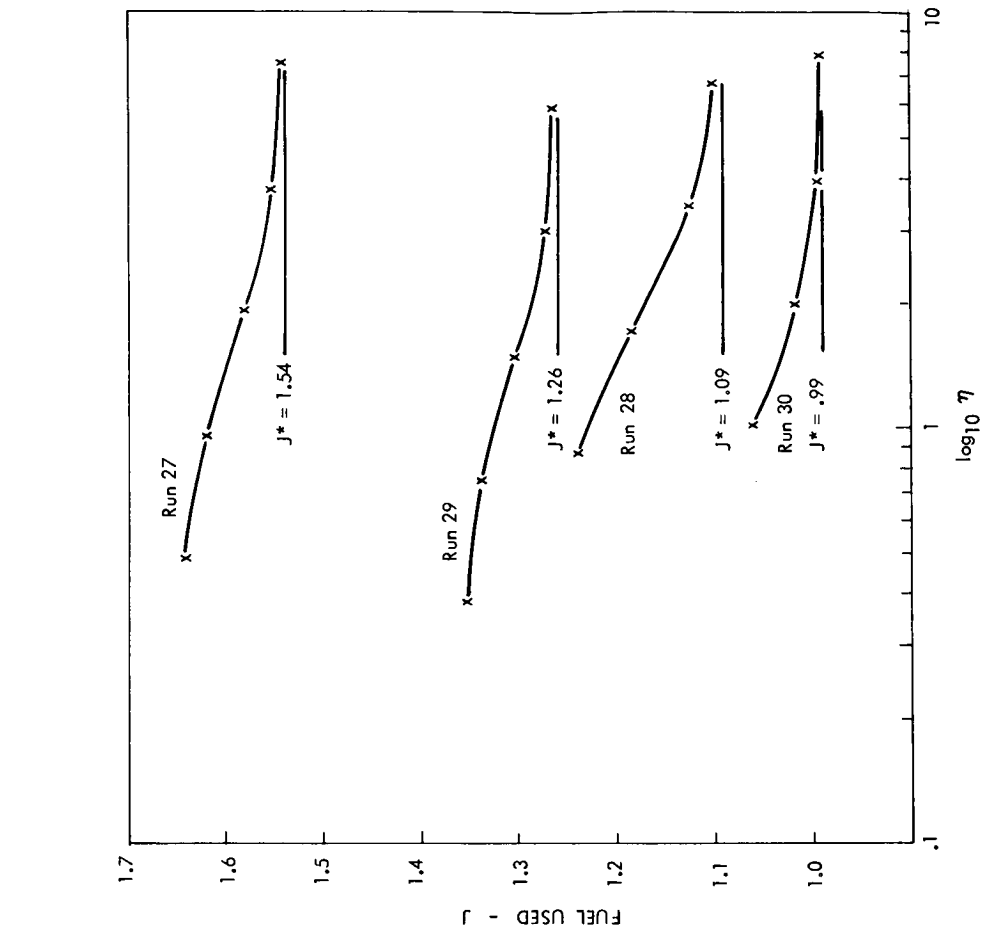


Fig. 5.24 Graph of Total Fuel Used vs. $\log_{10} \eta_k$ - Runs 26-29 For the Quadrupole Plant

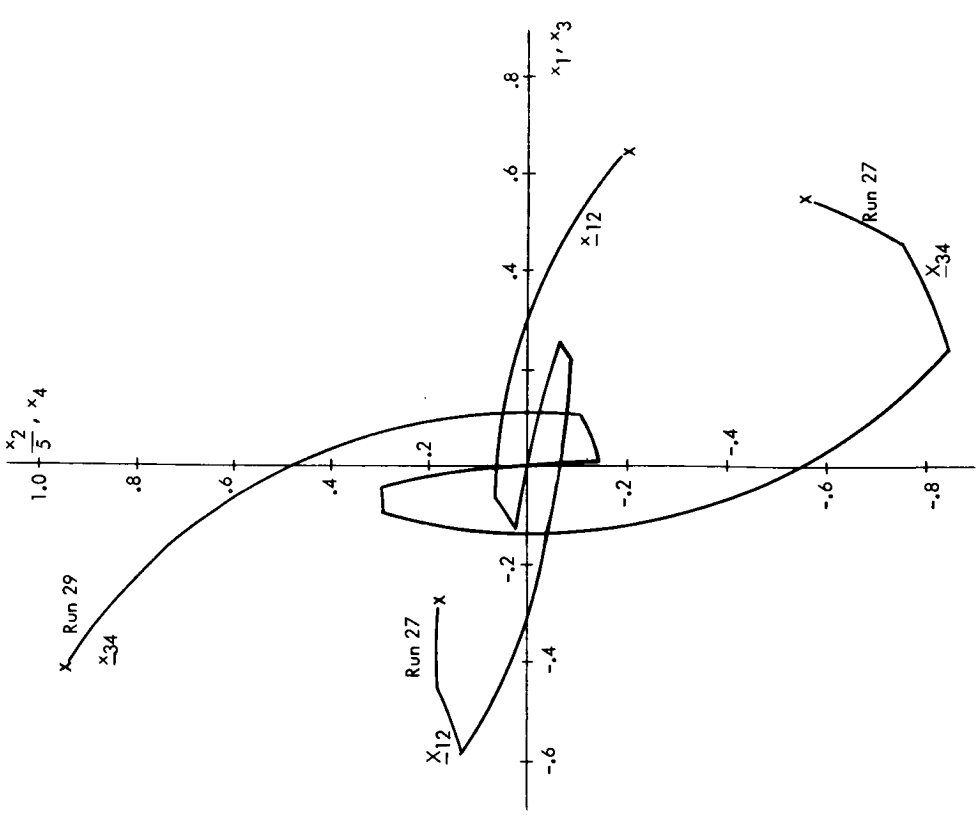


Fig. 5.23 State Space Trajectories for Fuel-Optimal Control of the Quadrupole Plant

In Fig. 5.25 the vectors $\underline{\pi}_{12}$ and $\underline{\pi}_{34}$ behave somewhat like the vectors $\underline{\pi}_k$ of the previous section, starting out in one direction until $\alpha \to 0$ and then converging toward $\underline{\pi}^*$. One difference is that the two directions are no longer nearly parallel in this run. Also the vector sequence $\{\underline{\pi}_k\}$ does not converge very close to the vector $\underline{\pi}^*$. Why it does not is still a mystery. With $M = 100$ the accuracy of the run is high, and the four vectors $\underline{q}(t_i)$ at the switch points do span the space R_4.

$$\underline{\text{Run 27}}: \quad \underline{\pi}^* = \begin{bmatrix} -5. \\ 0 \\ 0 \\ 5. \end{bmatrix} \quad ; \quad \text{see Eq. 5.9}$$

The designed control history for this run was 0, -1, 0, +1. The sequence of vectors $\{\underline{\pi}_k\}$ shown in Fig. 5.26 goes in one direction until $\alpha \to 0$, then it changes direction. In this run there seems to be no attempt to converge to the vector $\underline{\pi}^*$. Furthermore the vectors lie on curved (not straight) lines. Thus the effect of one degree of freedom in $\underline{\pi}^*$ seems strange in these two graphs, but might perhaps be plain if one graph could be plotted in a four dimensional space.

$$\underline{\text{Run 28}}: \quad \underline{\pi}^* = \begin{bmatrix} -1. \\ 0 \\ 0 \\ 3. \end{bmatrix} \quad ; \quad \text{see Eq. 5.9}$$

The designed control history for this run was -1, 0, +1. With the two degrees of freedom this gives, both the $\underline{\pi}_{12}$ and the $\underline{\pi}_{34}$ plots shown in Fig. 5.27 moved far out along straight lines. Actually the sequence first moved out, then part way back, and then out again, giving the best example found of the way in which the sequence $\{\underline{\pi}_k\}$ can depart from the vector $\underline{\pi}^*$. Even with this behavior all the operators converged.

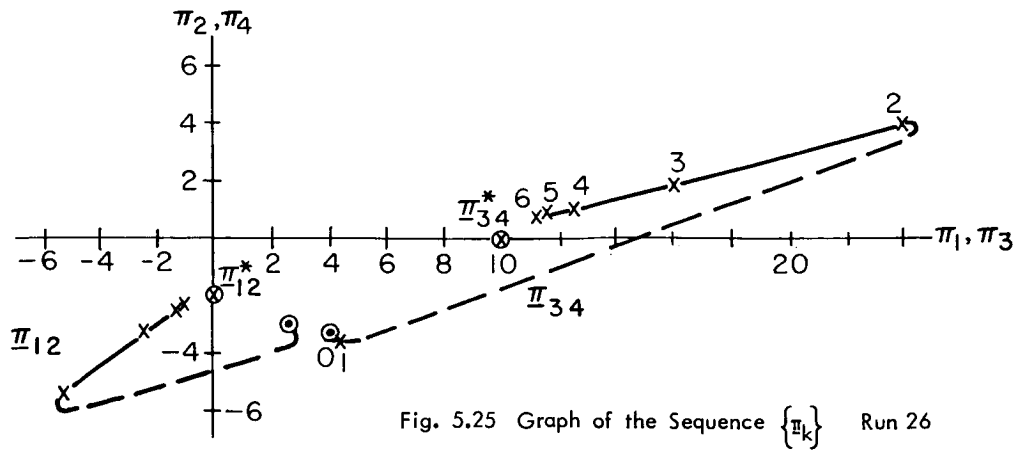


Fig. 5.25 Graph of the Sequence $\{\underline{\pi}_k\}$ Run 26

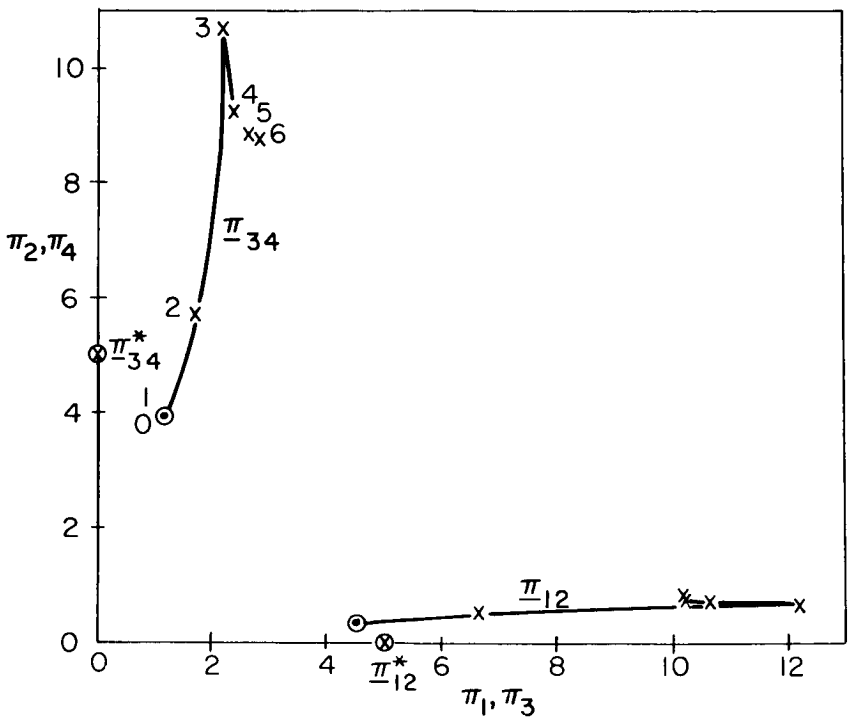


Fig. 5.26 Graph of the Sequence $\{\underline{\pi}_k\}$ Run 27

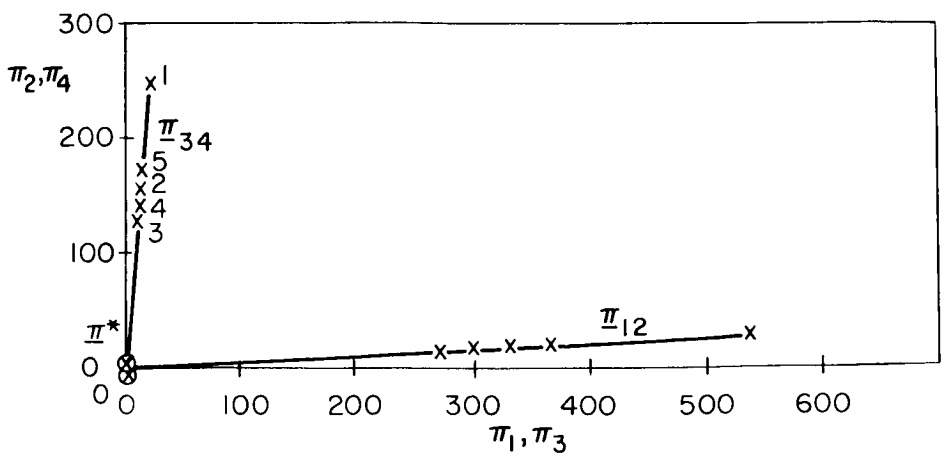


Fig. 5.27 Graph of the Sequence $\{\underline{\pi}_k\}$ Run 28

$$\underline{\text{Run 29}}: \ \underline{\pi}^* = \begin{bmatrix} 1. \\ .5 \\ \\ 0 \\ 0 \end{bmatrix} \ ; \quad \text{see Eq. 5.9}$$

The control history for this run was 0, +1, leaving three degrees of freedom. The result looks a bit like a combination of the last two runs. The sequence of vectors $\{\underline{\pi}_k\}$ starts off in one direction until $\alpha \to 0$, then proceeds along a straight line (but not one through the origin), and it does reverse direction along this straight line.

H. THE QUADRUPOLE OSCILLATOR PLANT

As a variation, the quadrupole plant is investigated with $\omega = 4$.

$$\underline{A} = \begin{bmatrix} 1 & 0 & & 0 \\ 0 & -1 & & \\ & & 0 & 4 \\ & 0 & -4 & 0 \end{bmatrix} \qquad \underline{b} = \begin{bmatrix} 1 \\ -1 \\ 0 \\ 1 \end{bmatrix} \qquad T = 2.0 \qquad (5.10)$$

The purpose was to extend the results of the previous section to a plant with a less symmetric arrangement of poles.

$$\underline{\text{Run 30}}: \ \underline{\pi}^* = \begin{bmatrix} 0 \\ 0 \\ \\ .5 \\ 1.5 \end{bmatrix}, \quad M = 100; \quad \text{see Eq. 5.10}$$

As expected, with the only nonzero entries in $\underline{\pi}^*$ occurring in the last two elements, the sequence of vectors $\{\underline{\pi}_k\}$ shown in Fig. 5.28 had most of its magnitude and variation in the last two elements (in $\underline{\pi}_{34}$).

The run consumed 58.1 seconds of computer time. If the trajectories in the state space are not needed (SSTRAJ is not used) 13.9 seconds of this can be saved.

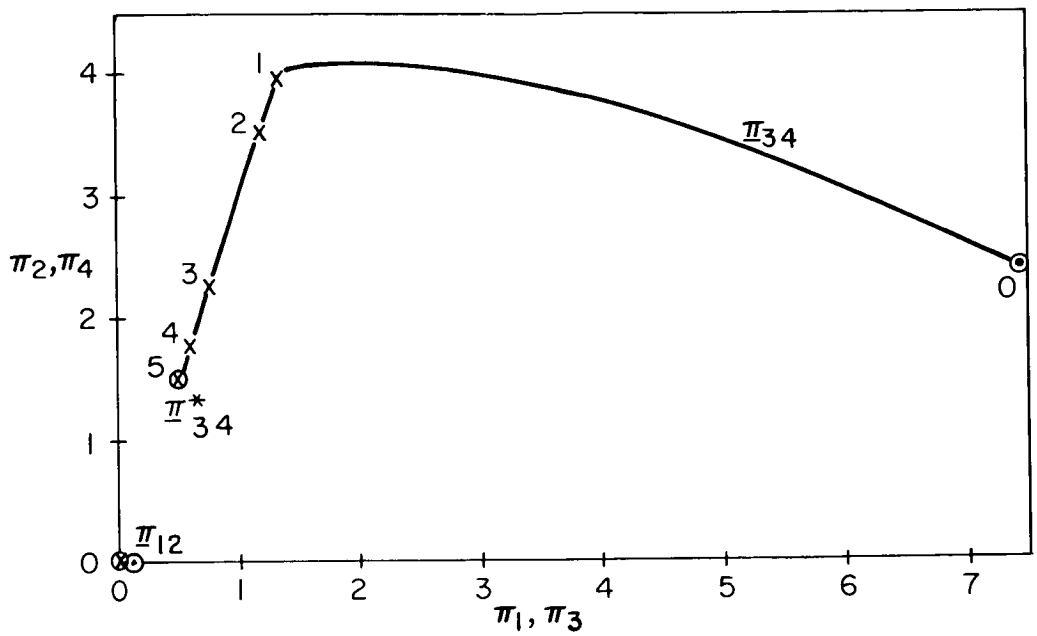


Fig. 5.28 Graph of the Sequence $\{\underline{\pi}_k\}$ Run 30

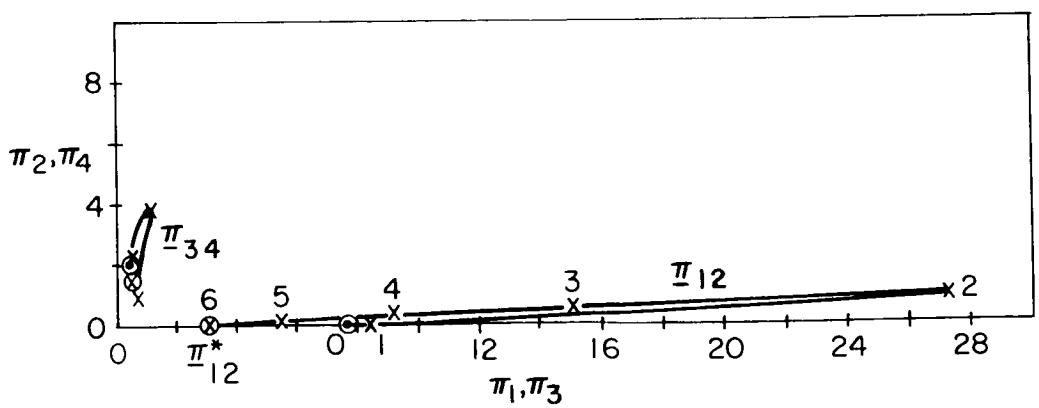


Fig. 5.29 Graph of the Sequence $\{\underline{\pi}_k\}$ Run 31

Run 31: $\underline{\pi}^* = \begin{bmatrix} -3. \\ 0 \\ \\ .5 \\ 1.5 \end{bmatrix}$ M = 100; see Eq. 5.10

The largest element of $\underline{\pi}^*$ is the first, and in Fig. 5.29 it is the elements π_1 in the sequence of vectors $\{\underline{\pi}_k\}$ which show the largest magnitudes.

As a side experiment, the same run was made with M = 25, 10, and 5. These runs consumed 22.9 seconds, 13.0 seconds, and 20.6 seconds of computer time respectively. With M = 100, 99.4 seconds of computer time were required, although in this case the state space trajectories were also computed.

Reducing M to 25 has a relatively small influence, while further reduction in M to 10 produced quite a marked error, of about 25 percent in the magnitude of the computed initial state $\underline{\xi}$.

The sequences of vectors $\{\underline{\pi}_k\}$ reinforce the above conclusion. For M = 100 and M = 25 the sequences were reasonably close to each other. For M = 10, two of the operators had to be redefined, giving a quite different sequence. Finally, with M = 5 the procedure could not be carried out successfully, in that the slope α was never reduced to zero.

I. THE TRIPLE OSCILLATOR PLANT

A sixth order plant, consisting of three oscillators (or an oscillator with three degrees of freedom) was examined briefly to try out the procedure on a larger order plant.

$$\underline{A} = \begin{bmatrix} 0 & 1 & & & & \\ -1 & 0 & & & & 0 \\ & & 0 & 2 & & \\ & & -2 & 0 & & \\ & & & & 0 & 4 \\ 0 & & & & -4 & 0 \end{bmatrix} \quad \underline{b} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} \quad T = 2\pi = 12.5664$$

The two runs performed were:

Run 33:

$$M = 100, \quad \underline{\pi}^* = \begin{bmatrix} 0 \\ .5 \\ 0 \\ .5 \\ 0 \\ 1. \end{bmatrix}$$

Run 34:

$$M = 40, \quad \underline{\pi}^* = \begin{bmatrix} 0 \\ 1. \\ 0 \\ 2. \\ 0 \\ 1. \end{bmatrix}$$

The results are similar to those for the double oscillator plant. Again, the sequence of vectors $\{\underline{\pi}_k\}$ lies on a straight line from the origin and converges to $\underline{\pi}^*$ (within the numerical accuracy used). Run 34 required a total of 27 iterations of Newton's method and consumed 109.5 seconds of computer time, including time to calculate the state space trajectories. Run 35 required a total of 30 iterations of Newton's method and consumed 40.4 seconds of computer time.

CHAPTER VI

DISCUSSION OF COMPUTER RESULTS

A. OVERALL CHARACTERISTICS

For almost all of the problems tried, a convergent sequence of approximations to the optimal control was produced. In those few cases where the slope α was not reduced to zero (and therefore the sequence of vectors $\{\underline{\pi}_k\}$ did not converge to a solution of the necessary conditions) it is suspected that no solution exists. However, see the section on accuracy.

One of the strengths of the method is its flexibility. Thus when Newton's method applied to a particular operator does not converge, another operator is defined until one is found for which convergence does result. As shown in Chapter III. this can always be done, and in such a way as to lead toward a solution of the necessary conditions of Pontryagin.

The present computer program allows for any form of linear, time-invariant plant (up to tenth order), and allows a choice of several constants and special feature subroutines. A number of possible extensions are described in the next chapter. The criterion for choosing the values of η_k and α_k appear to be efficient ones (they move toward the optimal solution rapidly without running into divergence of Newton's method too often), at least up through the sixth order example studied. As might be expected, the most difficult step is often the one in which the linear slope α is reduced to zero.

This program required relatively long running times on the digital computer; some of the most difficult runs consuming a minute or more on an IBM 7094. The times were made a little longer because the M.I.T. timesharing system was in operation when most of them were made, but this is thought to be a small factor. If computer time were a major concern, a faster program could be written, but this procedure was not designed for maximum speed of execution. Instead it was designed for reliability and to give some information about a set of suboptimal controls. In these areas the program did well, as

noted elsewhere in this chapter. Work could be done on a faster running program. One run was made with this idea in mind, and is reported in Section C of this chapter. An attractive technique for this purpose would be to vary the integration step size, by starting off with a small value of M (say M = 20, for instance). When the procedure runs into trouble or nears the end of the sequence of approximate operators, then the value of M would be increased.

The total fuel used was plotted against $\log_{10} \eta_k$, since the approximate control function $u_k(\cdot)$ is an exponential type of function. When the parameter η_k reaches a value of $\eta = 5$, usually the resulting cost is within 1% of J*, the optimal fuel cost. Note that the first operator usually has a cost 5 to 30% greater than J*, showing in practice the efficiency of the approximate controls u_k. The easiest runs, using the least fuel, are worst in this regard, since the optimal control u*(t) then chooses its on times more judiciously. Even in these cases the approximate controls were quite efficient in the experimental runs.

B. EFFECTIVENESS OF THE APPROXIMATE OPERATORS

In Chapter III the approximate operators were examined from a theoretical point of view. In Theorems 3.2 and 3.3 it was shown under certain assumptions that the approximate operator $T_k(\underline{\pi}^*)$ can be brought as close as desired to the true operator $T(\underline{\pi}^*)$, and in addition the solution vector $\underline{\pi}_k$ could be made as close to $\underline{\pi}^*$ as desired. The purpose of this section is to add a few practical comments based on the examples studied in Chapter V.

The sequence of solution vectors $\{\underline{\pi}_k\}$ generally had moved close to its final value when the parameter η_k had reached a value of $\eta = 2 - 5$. Then the accuracy used in the digital computations becomes an increasingly important factor in determining the distance between $\underline{\pi}_k$ and $\underline{\pi}^*$. As η_k increases beyond $\eta = 10$, usually only one iteration is enough to meet the criterion for convergence of Newton's method. If η_k is allowed to become very large (> 5000 in one case tried on the computer), another numerical difficulty may develop. Just why this happens is not known, and since it has no effect on the procedure in practice, it was not considered important.

Since only controls from the class of feasible controls were examined, the plant had a smoothing effect on any variations in the control function. This was true even for the plant with an unstable pole, and became especially true when the open loop equations were stable. Thus a small variation in the control function or in the vector $\underline{\pi}$ causes an even smaller variation in the state space trajectory.

C. ACCURACY

1 Integration Step Size

In the course of the numerical work, the question of accuracy came up repeatedly. One obvious source of inaccuracy occurred in carrying out the integrations numerically. A straightforward trapezoidal rule was used in approximating the integrals. More complex schemes could have been used, and the interval size can be made smaller in the trapezoidal rule; either of these approaches will result in greater accuracy when using the simple rational integrands required. However, there is always some loss of accuracy, and its effects need to be noted.

In Runs 11, 12, 19, and 20 a study was made of the effect of interval size or number of subdivisions in each integration. Another study was made in Run 32, using the quadrupole oscillator plant. Note that no matter how poor the integration scheme used; (1) the state space still gets to the origin at time $t = T$ for the model used, and (2) the run can always be repeated except for roundoff error in the digital computer.

As accuracy was decreased, the number of iterations of Newton's method required increased. The asymptotically quadratic convergence noted in Appendix B is for the exact Newton's method. With high accuracy runs this property was found in practice. However, as the accuracy goes down the iterations begin to stray and convergence occurs at a lower rate. Of course this can also cause Newton's method to diverge in a case where convergence would occur if higher accuracy were used

There is a relation between how difficult the problem is and the effect of inaccuracies. In Chapter VI, Section C.1 the difficulty of a problem is mentioned. In a qualitative way this depends on the distance

of the final state $\underline{\theta}$ from the boundary of the set of states reachable from the state $\underline{\xi}$ in T seconds. Thus for a given plant and time T, the closer the final state $\underline{\theta}$ is to the boundary of this set of reachable states, the more difficult the problem is.

As the accuracy decreases the problem becomes more difficult. Thus in Run 32, the accuracy was finally reduced (M = 5) to the point where the problem became too difficult to solve, even though there was no special difficulty with the problem when using greater accuracy (M = 100 or 25). Usually this happens because the inverse of the first derivative operator fails to exist.

This point is illustrated qualitatively in Fig. 6.1 using two dimensions. The plant, the final state $\underline{\theta}$ and the final time T are assumed

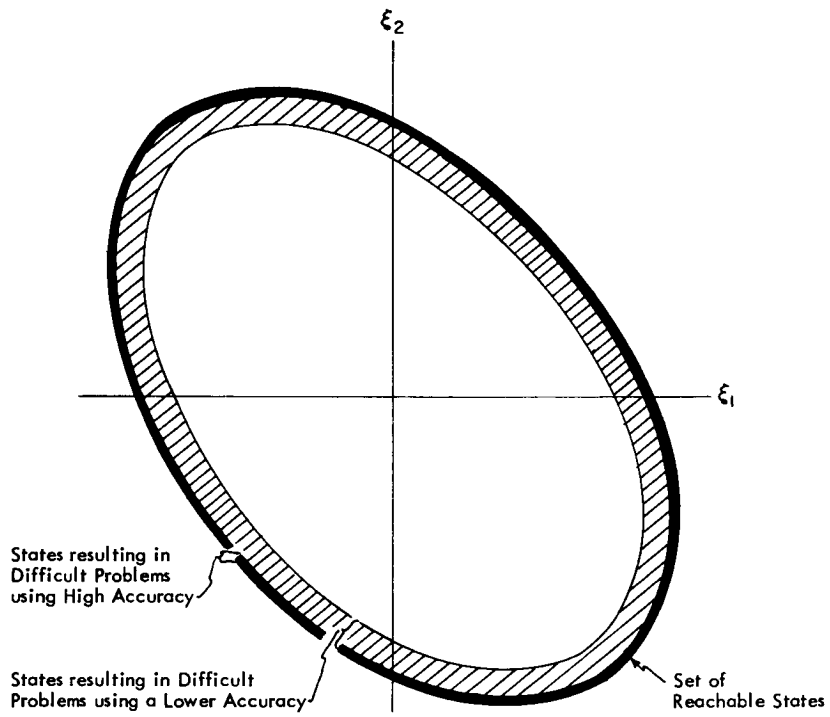


Fig. 6.1 Diagram of Initial States $\underline{\xi}$ and Problem Difficulty

fixed. Then there is a set of initial states $\underline{\xi}$ for which the problem is easy, one for which the problem is difficult, and another set for which the problem is impossible. As shown in the sketch of Fig. 6.1, using high accuracy results in a narrow band of initial states $\underline{\xi}$ for

which the problem is difficult. With reduced accuracy this band of initial states leading to difficult problems becomes much wider.

Of course the digital computer has a limited accuracy which introduces roundoff errors, so there is no hope of getting rid of this region of difficult problems completely. In addition, the nature of the procedure itself must introduce some difficulty. These two effects are difficult to separate.

A great deal of work could be done in trying to evaluate the size and effect of the difficult region. However, it should be sufficient to know that the region can be made small (or narrow). This is indicated, for instance, by Runs 15 - 18, which did not lead to any difficulty even as the vector $\underline{\pi}^*$ was doubled.

If the procedure fails to work on a given problem, the first remedy to try is to increase the accuracy. If the accuracy was already known to be high or increased accuracy does not result in a solution, then in most cases the problem will be found to be impossible. One way to check this, of course, would be to compute the time optimal solution and compare the minimum time T^* with the given time T; if $T^* > T$ the problem has no solution (is impossible).

Another effect of the numerical integration is to make the optimal costate initial condition vector $\underline{\pi}^*$ nonunique. Once a set of mesh points is chosen, then any vector $\underline{\pi}$ which yields the same value of the control (+1, 0, or -1) at each of these points can serve as an optimal solution. There is generally a compact set of vectors W satisfying this condition. The sequence of vectors $\{\underline{\pi}_k\}$ converges toward a sort of average vector in this set W. Each vector $\underline{\pi}_k$ is still unique if the conditions of Chapter III (Assumptions 3.1 - 3.3, and Lemma 3.1) are fulfilled.

As a result of this the sequence of vectors $\{\underline{\pi}_k\}$ converges to a solution of the TPBVP, a vector $\underline{\pi}^* \in W$, but not necessarily to the vector $\underline{\pi}^*$ that was used in choosing the state initial condition $\underline{\xi}$. This can be seen, for instance, in Figs. 5.20 - 5.22, where the sequence of vectors $\{\underline{\pi}_k\}$ does not converge exactly to the given vector $\underline{\pi}^*$. Also, as the mesh is made more coarse, the sequence $\{\underline{\pi}_k\}$ may converge to a point further out from the given vector $\underline{\pi}^*$.

2. Other Approximations

In addition to the integration step size, chosen by picking the number of mesh points M, there are several other computer approximations affecting the accuracy.

At the beginning of the program the matrix exponential $\underline{e}^{-\underline{A}t}$ is computed from the series definition, and the process is terminated by choosing a constant EPMTX, as shown in Chapter IV. Any error here will show up at every later step of the program, and will be difficult to find without a complete rerun of the problem. Also, this calculation is only performed once. So a high accuracy is normally used (EPMTX = 10^{-6}), which insures that $\underline{e}^{-\underline{A}t}$ will be accurate down to almost the level of roundoff errors.

For each approximate operator $T_k(\underline{\pi})$ a constant called EPS is used to decide when Newton's method has converged. Poor accuracy here means each solution vector $\underline{\pi}_k$ will be in error. This can lead to divergence in trying to solve the next operator $T_{k+1}(\underline{\pi})$. In an extreme case it could even lead to acceptance of a spurious solution for $\underline{\pi}_k$. In most problems the quadratic convergence of Newton's method keeps the error smaller than EPS. A quick calculation shows that using the normal value of EPS(EPS = 10^{-3}) with quadratic convergence yields an error in $\| \underline{\pi}_k \|$ of 10^{-5} or even 10^{-6} if Newton's method is carried out exactly. With integration and roundoff errors (and also sometimes in difficult problems) this error may be larger. An experiment was made with EPS = 10^{-1} (in the run reported below) resulting in a little jitter in the sequence $\{\underline{\pi}_k\}$ but no other noticeable bad effects.

The sequence of approximate operators $T_k(\underline{\pi})$ is terminated when η_k exceeds the constant AMAX. If AMAX is chosen too small, the sequence will terminate before getting close to $\underline{\pi}^*$. Making AMAX too large just leads to extra computations, although Runs 1 and 3 indicated that numerical troubles could result in extreme cases. By the time η_k reaches a value of 2 - 5, the sequence $\{\underline{\pi}_k\}$ has pretty well converged, so taking AMAX = 10 seems to be a good value.

Finally, it should be noted that all of the above approximations are involved in the trade-off between computer time and accuracy. For example, in Run 30 the recommended values were used, and the

run consumed 58.1 seconds of computer time. Then the run was repeated using

$$M = 25$$

$$EPMTX = 10^{-6}$$

$$EPS = .1$$

$$AMAX = 2.$$

With these changes the run consumed only 8.0 seconds of computer time, and the results only deteriorated a little from an engineering point of view (final vector $\underline{\pi}_k$ changed by about 15 percent).

D. STRAIGHT LINE BEHAVIOR OF $\{\underline{\pi}_k\}$

When the system (open loop) poles lie on the imaginary axis, the sequence of vectors $\{\underline{\pi}_k\}$ was found to lie on a straight line through the origin. This is true also for the fourth order and sixth order examples in the vector spaces R_4 and R_6, respectively. No reason has been found for this. It is surmised (unproven) that the straight line behavior will hold for any sequence of control approximations u_k having symmetry about the origin, acting on a conservative system in a fixed time control problem.

A suboptimal control system can be designed for a linear, conservative system based on this straight line property: Suppose the state of the system is given at time $t = 0$, and it is required to guide the system open loop for T seconds, until the next fix on the state will be given.

If the control were linear with slope a, then from Change 2, of Chapter II, the costate initial condition would be,

$$\begin{aligned} \underline{\pi} &= \frac{1}{a} \underline{W}^{-1}(T) [\underline{\xi} - e^{-AT} \underline{\theta}] \\ &= \frac{1}{a} \underline{W}^{-1}(T) \underline{\zeta} \end{aligned} \tag{6.1}$$

The key of this design is to use the costate resulting from Eq. 6.1 with the optimal control function $-dez(\cdot)$. Because of the straight line behavior of $\{\underline{\pi}_k\}$ there is some slope a for which this gives the

optimal result. The effect of α is to determine $\|\underline{\pi}\|_2$, the length of the vector $\underline{\pi}$. We set

$$\begin{aligned} u(t) &= -\text{dez}[\underline{q}'(t)\underline{\pi}] \\ &= -\text{dez}[\tfrac{1}{\alpha}\underline{q}'(t)\underline{W}^{-1}(T)\underline{\zeta}] \end{aligned} \tag{6.2}$$

It remains to choose the constant α. Clearly, the smaller α is the closer the control to a time optimal one; the larger α is the slower the control but also the more efficient in its use of fuel. The safest way to pick α is by test of the system under field conditions.

With certain systems it may be possible to design a rule for choosing α. For instance, in the single oscillator control problem to the origin, the time optimal control reduces $\|\underline{x}(t)\|_2$ by about two units every π/ω seconds, so the minimum time T^* is approximately,

$$T^* \approx \frac{\pi}{2\omega}\|\underline{\zeta}\|_2 \tag{6.3}$$

One way to choose α would be to give the control argument a magnitude based on the ratio T^*/T. For example, let the magnitude be $1.0 + T^*/T$. Then as $T^*/T \to 0$ the control effort also goes to zero. For clarity this will be done in two steps. First, change the magnitude of the argument to one.

$$\frac{\underline{q}'(t)\underline{W}^{-1}(T)\underline{\zeta}}{|\underline{q}'(0)\underline{W}^{-1}(T)\underline{\zeta}|}$$

Then multiplying by the chosen magnitude yields the desired control.

$$u(t) = -\text{dez}\left[(1 + \frac{\pi}{2\omega T}\|\underline{\zeta}\|_2)\ \frac{\underline{q}'(t)\underline{W}^{-1}(T)\underline{\zeta}}{|\underline{q}'(0)\underline{W}^{-1}(T)\underline{\zeta}|}\right] \tag{6.4}$$

E. THE CONVERGENCE THEOREM OF KANTOROVICH

In Appendix B the basic theorem on the convergence of Newton's method is presented. It is the keystone of the theoretical part of this thesis, but proved to be of little use in making practical estimates of the region of convergence. Some reasons why this is so are given below.

The main point is that the theorem provides only a sufficient condition for convergence. This is fine for proving other theorems and designing a safe sequence of approximate operators $\{T_k(\underline{\pi})\}$, where a guaranteed convergence is desired.

For the design of an efficient sequence of approximate operators $\{T_k(\underline{\pi})\}$ it would help to have a necessary condition for convergence. Example B.3 of Appendix B shows that the sufficient condition can also be necessary. But it generally is not, and may, in fact, be very far from necessary as is shown in Example B.1.

In the numerical computations two approximations were needed, both of which made the sufficient condition for convergence even further from being necessary. First, the required norm of the second derivative operator has to be bounded by means of an inequality, as shown in Chapter II, Section F. There are cases for which this upper bound is exact, but it generally overestimates the norm.

Second, the norm of the second derivative operator must be evaluated over a certain region of the vector space of the vectors $\underline{\pi}$, and the maximum value taken. This was found to be very difficult, and a rough upper bound was used. The resulting values for this norm turned out to be very large especially for large values of n, so an estimate of the norm was made for comparison. In the estimate, instead of evaluating the norm for all possible values of $\underline{\pi}$ in a region around the starting guess $\underline{\pi}_0$, the norm is only evaluated at the vector $\underline{\pi}_0$. The estimated parameter is called h_1, to distinguish it from the guaranteed convergence parameter h.

Some numerical results are shown in Fig. 5.5, Chart 5.18, and Fig. 5.19. The figures are plotted with the number of iterations of Newton's method required for convergence as the abcissa. At the right-hand end of the abcissa a space is reserved for the divergence of Newton's method. Note that the parameters h and h_1 usually increase as the number of iterations increases.

The parameters h and h_1 are almost always larger than the maximum value of 1/2 allowed by the theorem. Yet, Newton's method converged in most of the cases shown. The parameter h went as high as $.4 \times 10^6$ with convergence still resulting. The estimated parameter h_1 is about one order of magnitude smaller than h

in Fig. 5.5. In Chart 5.18 it varies from about 1-1/2 to 3-1/2 orders of magnitude smaller than h, perhaps due to the unstable root.

For a given plant the parameter h_1 could be used to make a rough estimate of whether Newton's method would converge and how many iterations would be required. Between two different plants the prediction is not as reliable, especially as the number of dimensions of the state space changes.

It was suggested that the parameter h be used, somehow, in the practical design of the procedure. In retrospect, this idea seems limited because:

1. The parameter attains such large values in comparison with the value 1/2 given in Theorem.
2. It is not too reliable for use on many different plants.
3. It requires quite a bit of computer time to compute it.

F. AN APPLICATION

Two physical problems are suggested as examples of how the double oscillator plant might occur. The first is a simple mechanical device consisting of two single degree of freedom pendulums of different lengths hanging from a common support as in Fig. 6.2. If the

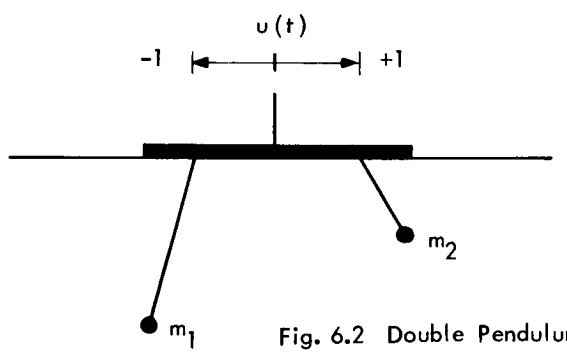


Fig. 6.2 Double Pendulum Example

support can be moved a limited amount as shown, then the problem would be to get both pendulums stopped in T seconds.

Another problem is the small angle attitude control of an earth (or other planet) satellite in a circular (or near circular) orbit. There

are three axes of rotation, body centered as shown in Fig. 6.3. Differential equations for the small angle motions have been derived by DeBra[23] and others, and are shown below.

$$I_1(\ddot{\theta}_1 - f\dot{\theta}_2) + f(I_3 - I_2)(\dot{\theta}_2 + f\theta_1) = T_1 \qquad \text{Yaw}$$

$$I_2(\ddot{\theta}_2 + f\dot{\theta}_1) + f(I_1 - I_3)(\dot{\theta}_1 - f\theta_2) = -3f^2(I_3 - I_1)\theta_2 + T_2 \qquad \text{Roll}$$

$$I_3\ddot{\theta}_3 = -3f^2(I_2 - I_1)\theta_3 + T_3 \qquad \text{Pitch}$$

where f is the orbital frequency. (6.5)

The pitch equations are independent of the others, making control about that axis a separate, simpler problem. A separate control

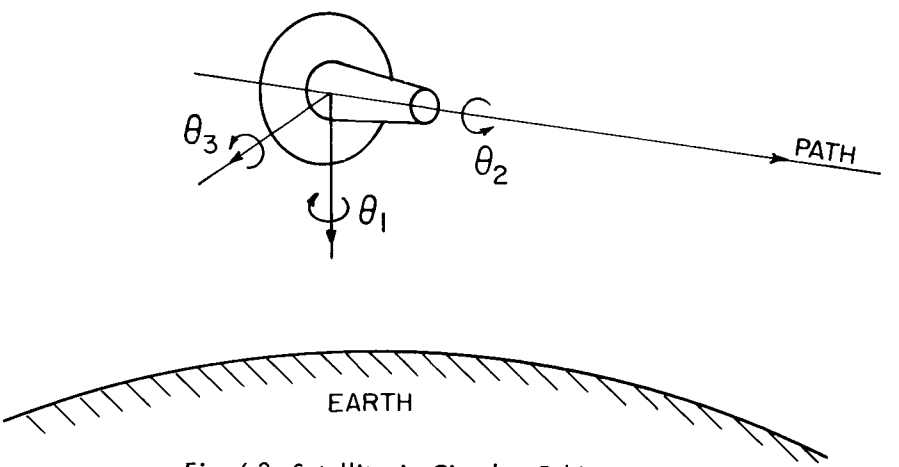


Fig. 6.3 Satellite in Circular Orbit

thruster might be used for this axis. Solutions for the single oscillator problem that results will not be considered here.

The remaining equations are now written in matrix form.

Let $\dfrac{T_1}{f^2 I_1} = u_1$ $\qquad$ $\dfrac{T_2}{f^2 I_2} = u_2$

$\dfrac{I_3 - I_2}{I_1} = a_1$ $\qquad$ $\dfrac{I_3 - I_1}{I_2} = a_2$

and make a change in the time scale.

$$\sigma = ft \qquad \text{so} \qquad \frac{d\underline{x}}{d\sigma} = \underline{x}' = \frac{1}{f}\frac{d\underline{x}}{dt}$$

Let

$$\underline{x} = \begin{bmatrix} \theta_1 \\ \theta_1' \\ \theta_2 \\ \theta_2' \end{bmatrix}$$

Then the yaw and roll equations become

$$\underline{x}' = \underline{A}\,\underline{x} + \underline{u}$$

or

$$\underline{x}' = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -a_1 & 0 & 0 & 1-a_1 \\ 0 & 0 & 0 & 1 \\ 0 & a_2^{-1} & -4a_2 & 0 \end{bmatrix} \underline{x} + \begin{bmatrix} 0 \\ u_1 \\ 0 \\ u_2 \end{bmatrix} \tag{6.6}$$

For further transformation, the natural frequencies will be needed

$$\det\left[\underline{A} - \omega\underline{I}\right] = \omega^4 + \left[1 + 3a_2 + a_1a_2\right]\omega^2 + 4a_1a_2 = 0 \tag{6.7}$$

Let ω_1 and ω_2 be the solutions of Eq. 6.7 with $\omega_2 \geq \omega_1$. In order to have the form used in the computer examples (one frequency equal to 1.0) a second change is made in the time variable.

Let

$$\tau = \omega_1\sigma = f\omega_1 t \tag{6.8}$$

so

$$\frac{d\underline{x}}{d\tau} = \dot{\underline{x}} = \frac{1}{\omega_1}\frac{d\underline{x}}{d\sigma} = \frac{1}{f\omega_1}\frac{d\underline{x}}{dt}$$

Finally, a change in the state variables, called a similarity transformation is made in order to decouple the two natural modes of vibration. Suppose, for instance, that only one control is available, exerting a thrust about the roll (θ_2) axis. Then the appropriate transformation would be

$$\underline{Y} = \underline{P}\underline{X}$$

where

$$\underline{P} = f^2 I_1 \begin{bmatrix} 0 & \frac{\omega_1(a_1-\omega_2^2)}{a_1(1-a_1)} & \frac{4a_2}{\omega_1} & 0 \\ \frac{a_1-\omega_2^2}{1-a_1} & 0 & 0 & 1 \\ 0 & \frac{\omega_2(a_1-\omega_1^2)}{a_1(1-a_1)} & \frac{4a_2}{\omega_2} & 0 \\ \frac{a_1-\omega_1^2}{1-a_1} & 0 & 0 & 1 \end{bmatrix} \qquad (6.9)$$

The resulting vector equation is then,

$$\dot{\underline{y}} = \begin{bmatrix} 0 & 1 & & \\ -1 & 0 & & \\ & & 0 & \frac{\omega_2}{\omega_1} \\ & & -\frac{\omega_2}{\omega_1} & 0 \end{bmatrix} \underline{y} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} T_2 \qquad (6.10)$$

where T_2 is the control torque about the roll axis.

In evaluating the difficulty of controlling a given plant three factors are important. The first is the ratio of the two frequencies ω_2 and ω_1 from Eq. 6.7. As this ratio approaches 1.0, the plant becomes more difficult to control. The second is the magnitude of the entries in the similarity transformation matrix, Eq. 6.9, which determines by how much a given initial condition $\underline{\xi}$ on the state vector is magnified. Third is the time scaling in Eq. 6.8 which determines by what factor the given terminal time T is scaled. For any given problem, all three of these factors must be taken into account.

The control task can be made easier by careful design of the satellite's principle moments of inertia. I_3 should be greater than I_2 and

also greater than I_1 in order to have two oscillatory modes. Otherwise the roots become real, and one of the pair will be unstable. Also, I_3 should not be too close to the sum of I_1 and I_2, in order to avoid large entries in the similarity transformation matrix 6.9.

CHAPTER VII

GENERAL DEVELOPMENT OF METHOD

In this chapter the fixed time, fixed terminal state optimization problem is examined. The steps undertaken are similar to those in Chapter II, but here the more general relations are shown. A restricted version of the problem is then defined, and the simplifications that result are pointed out. The purpose is to show in some detail how the approach of Chapter II can be applied to a more complex problem, and to point out some of the difficulties that result. Some alternate approaches are pointed out. Chapter VIII shows how this approach can be extended to some other classes of problems.

A. PROBLEM 2

Given:

(a) A system (plant) described by the nonlinear vector differential equation, the state equation.

$$\dot{\underline{x}}(t) = \underline{f}(\underline{x}(t), \underline{u}(t), t) \tag{7.1}$$

(b) A fixed time interval

$$t \epsilon [t_0, t_1]$$

(c) Initial and terminal boundary conditions on the state vector.

$$\underline{x}(t_0) = \underline{\xi}$$

$$\underline{x}(t_1) = \underline{\theta}$$

(d) The control variable must satisfy a constraint.

$$\underline{u}_{(t_0, t_1]} \epsilon U_{(t_0, t_1]} \tag{7.2}$$

In most cases the set U_t of allowable controls at time t will be bounded and convex. It is also assumed that the control function u(t) is piecewise continuous on $(t_0, t_1]$.

(e) A cost functional in integral form.

$$J(\underline{x},\underline{u}) = \int_{t_0}^{t_1} L(\underline{x}(\tau), \underline{u}(\tau), \tau)\, d\tau \qquad (7.3)$$

Then:

It is desired to find a control $\underline{u}^*(t)$ that:

(a) Satisfies the constraint 7.2.

(b) Transfers the system 7.1 from the initial state $\underline{\xi}$ at time $t = t_0$ to the terminal state $\underline{\theta}$ at time $t = t_1$.

(c) Minimizes the cost functional 7.3.

This set of conditions will be called Problem 2. It is in the form of a problem of Lagrange in the calculus of variations, since the functional to be minimized consists of an integral.

If the given terminal state $\underline{\theta}$ happens to be an equilibrium point, of the state equation 7.1, then Problem 2 is called a regulator problem. In particular, there usually exists at least one linear transformation of the state variables $\underline{x}(t)$ which makes the state equations homogeneous for $\underline{u}(t) = 0$. Then $\underline{x} = \underline{0}$ is an equilibrium point, and the terminal boundary condition $\underline{x}(t_1) = \underline{0}$ leads to a regulator problem.

B. THE TWO-POINT BOUNDARY VALUE PROBLEM*

The relations deduced by applying Pontryagin's Minimum Principle to Problem 2 are summarized below. See Appendix A, Section 7 for a statement of the Minimum Principle.

$$H(\underline{x},\underline{u},\underline{p},t) = L(\underline{x},\underline{u},t) + \underline{p}'(t)\underline{f}(\underline{x},\underline{u},t) \qquad (7.4)$$

$$\dot{\underline{x}}(t) = -\frac{\partial H}{\partial \underline{p}(t)} = \underline{f}(\underline{x},\underline{u},t)$$

---

* The two-point boundary value problem will be abbreviated to TPBVP.

$$\dot{\underline{p}}(t) = -\frac{\partial H}{\partial \underline{x}(t)} = -\frac{\partial L}{\partial \underline{x}(t)} - \left(\frac{\partial \underline{f}}{\partial \underline{x}(t)}\right)' \cdot \underline{p}(t) \qquad (7.5)$$

or

$$\dot{\underline{p}}(t) = \underline{g}(\underline{x}, \underline{u}, \underline{p}, t) \qquad (7.5a)$$

$$\underline{x}(t_0) = \underline{\xi} \qquad (7.6)$$

$$\underline{x}(t_1) = \underline{\theta} \qquad (7.6a)$$

and the relation for the optimal control

$$H(\underline{x}^*, \underline{u}^*, \underline{p}^*, t) \leq H(\underline{x}^*, \underline{u}, \underline{p}^*, t) \qquad \text{for all } \underline{u} \in U_t \qquad (7.7)$$

Comment 7:1 As in Chapter II, knowledge of $\underline{\pi}^*$, the costate initial condition vector is sufficient to reduce the TPBVP to an initial value problem (which requires 2n straightforward integrations).

Relation 7.7 may or may not have an explicit solution for $\underline{u}^*(t)$ in terms of $\underline{x}^*(t)$ and $\underline{p}^*(t)$. In many cases of practical interest it does have one. If it does not, the operations shown below can still be carried out, but relation 7.7 has to be carried along as an extra equation. In order to avoid this and keep the exposition simple, Assumption 7.1 is made.

Assumption 7:1 Relation 7.7 has an explicit solution, written as

$$\underline{u}^*(t) = \underline{v}[\underline{x}^*(t), \underline{p}^*(t), t] \qquad (7.8)$$

and well defined on the product space $R_n \times R_n \times [t_0, t_1]$, except possibly for a set of measure zero. See Chapter III for a discussion of the singular control problem.

Now the control terms can be eliminated from the state and costate Eqs. 7.1 and 7.5a, using relation 7.8.

$$\dot{\underline{x}}(t) = \underline{f}(\underline{x}, \underline{v}[\underline{x}, \underline{p}, t], t) \qquad (7.9)$$

$$\dot{\underline{p}}(t) = \underline{g}(\underline{x}, \underline{v}[\underline{x}, \underline{p}, t], \underline{p}, t) \qquad (7.10)$$

Equations 7.9 and 7.10 with the boundary conditions 7.6 and 7.6a constitute the TPBVP. For simplicity, these two sets of n equations each, are combined into one nonlinear vector equation of dimension 2n.

$$\underline{y}(t) = \begin{bmatrix} \underline{x}(t) \\ \text{----} \\ \underline{p}(t) \end{bmatrix} \tag{7.11}$$

and

$$\dot{\underline{y}}(t) = \underline{h}(\underline{y}(t), t) \tag{7.12}$$

A solution of the TPBVP is called an extremal solution of the original problem.

C. INTEGRAL EQUATION FORM

Because of the general nonlinear character of the Eq. 7.12, they cannot be handled and transformed with the assurance possible in Chapter II, Section C. Nonetheless, certain formal relations can be written which are valid in many cases. The final integral relation is of the Fredholm type, but the transformation will be made in two steps, to indicate more clearly the difficulty involved.

First consider Eq. 7.12 as an initial value problem, with

$$\underline{y}(t_0) = \begin{bmatrix} \underline{\xi} \\ \text{----} \\ \underline{\pi} \end{bmatrix} = \underline{\zeta} \tag{7.13}$$

Then if a solution exists, it must satisfy the Volterra integral equation

$$\underline{y}(t) = \underline{\zeta} + \int_{t_0}^{t} \underline{h}(\underline{y}(\tau), \tau)\, d\tau \tag{7.14}$$

A Lipschitz condition at each time t and an absolute integrability condition are sufficient to guarantee a unique solution $\underline{z}(\underline{\zeta}, t)$ to Eq. 7.14 (for a simple exposition see e.g., Tricomi[92], pages 41-47). This can be written as the result of successive approximations using Picard's method.

Let

$$\underline{z}_0(\underline{\zeta}, t) = \underline{\zeta} \tag{7.15}$$

and

$$\underline{z}_k(\underline{\zeta}, t) = \underline{\zeta} + \int_{t_0}^{t} \underline{h}(\underline{z}_{k-1}(\underline{\zeta}, \tau), \tau)\, d\tau \tag{7.16}$$

Then under the conditions above

$$\lim_{k \to \infty} \{\underline{z}_k(\underline{\zeta}, t) - \underline{\zeta} - \int_{t_0}^{t} h(\underline{z}_k(\underline{\zeta}, \tau), \tau)\, d\tau\} = \underline{0}$$

for all $t \in [t_0, t_1]$

and $\underline{z}(\underline{\zeta}, t)$ is defined by

$$\underline{z}(\underline{\zeta}, t) = \lim_{k \to \infty} \underline{z}_k(\underline{\zeta}, t) \tag{7.17}$$

In order to meet the terminal boundary condition 7.6a, the solution 7.17 must satisfy

$$[\underline{I} \vdots \underline{0}]\, \underline{z}(\underline{\zeta}, t_1) = \underline{\theta} \tag{7.18}$$

The operator $T(\underline{\pi})$ is then defined by

$$T(\underline{\pi}) = [\underline{I} \vdots \underline{0}]\, \underline{z}(\underline{\zeta}, t_1) - \underline{\theta} \tag{7.19}$$

Finally, Problem 2 has now been replaced by the problem of choosing the costate initial condition vector $\underline{\pi}$, such that

$$T(\underline{\pi}) = \underline{0} \tag{7.20}$$

D. SEQUENCE OF APPROXIMATE OPERATORS

The same kinds of changes made in Chapter II, Section D, can be made here. However, there is now a much greater choice in the way the changes are made. There are k control variables, the control variables may enter the state equations in nonlinear ways, and there may be nonlinearities among the state variables. Approximations are considered for all possible nonlinearities.

Change 1. Each control variable $u_i(\cdot)$ is a scalar function of the costate vector, the state vector and time. The form of the argument may be much more complex than the inner product found in Chapter II.

Usually each control variable is a piecewise continuous function of its argument. Then the theory of approximations states that a sequence of approximate functions can be found that converges (at least pointwise) to the given control function. There are d control variables and hence d approximating controls.

Change 2. The control variables may enter Eq. 7.12 in nonlinear ways. Also there may be nonlinearities in Eq. 7.12 not involving the control variables. In order to be sure of finding a sequence such that Newton's method converges when applied to each member sequentially, all the nonlinearities must be approximated in the sequence. If, however, it is suspected that some of the nonlinearities have little effect, it may save time to try Newton's method without approximating these.

Change 3. In order to start the procedure a linear (or nearly linear) operator should be used. This should be constructed so that it is a natural result of "spreading out" the nonlinearities until they approach linearity.

All these changes lead to a sequence of approximate operators, the k^{th} one of which is denoted by

$$T_k(\underline{\pi}) \tag{7.12}$$

Each Approximate operator has some approximate functions (changes) of type 1, denoted by subscripts

$$\eta, \theta, i, \ldots\ldots$$

and some of type 2, denoted by subscripts

$$\alpha, \beta, \gamma, \ldots\ldots$$

Just as in Chapter II, the procedure is to start with a linear (or nearly linear) operator and proceed by steps toward the given operator $T(\underline{\pi})$. It is desirable to have the approximate functions of type 2 become exact early in the sequence if possible. If a subscript zero stands for a linear approximation, and the subscript ∞ stands for the exact function, then a typical sequence might have the subscripts below:

Operator Number:	$\underline{0}$	$\underline{1}$	$\underline{2}$	$\underline{3}$	.....	$\underline{\ell-1}$	$\underline{\ell}$
	0	η_1	η_2	η_3	.....	$\eta_{\ell-1}$	∞
	0	θ_1	θ_2	θ_3		$\theta_{\ell-1}$	∞
	0	i_1	i_2	i_3		$i_{\ell-1}$	∞
	0	α_1	α_2	∞		∞	∞
	0	β_1	β_2	β_3		∞	∞
	0 (linear)	∞	∞	∞		∞	∞ (exact)

However, there is a wide range of variation possible in handling a general nonlinear problem, and the above schema is only a suggested approach. The size of the steps to be made is a matter of experience. There is, of course, a safety feature:

1. Suppose that step k was too large, so that Newton's method diverges on the $k^{\underline{th}}$ operator.
2. Then simply choose a smaller step and redefine the $k^{\underline{th}}$ operator.
3. Now use the solution vector $\underline{\pi}_{k-1}$ again to start Newton's method on the new $k^{\underline{th}}$ operator.

E. APPLYING NEWTON'S METHOD

Newton's method is to be applied to a typical operator $T_k(\underline{\pi})$. As in Chapter II this amounts to linearizing the operator T_k about the present iterate $\underline{\pi}^i$, and solving for the zero of the resulting linear operator. The recursive relation is the same as Eq. 2.21.

$$\underline{\pi}^{i+1} = \underline{\pi}^i - [T_k^{(1)}(\underline{\pi}^i)]^{-1} T_k(\underline{\pi}^i) \qquad (7.22)$$

The definitions of a derivative and an inverse in function space, given in Appendix B, are to be applied to evaluate the derivative of the operator T_k and the inverse of the derivative operator. This may be difficult to do. Two kinds of behavior can arise:

1. If the initial value problem 7.14 leads to an analytic or closed form solution for $\underline{y}(t)$ for all values of $\underline{\zeta}$ then the operator $\underline{T}_k(\underline{\pi})$ can be expressed in analytic or closed form. It may then be possible to express the first derivative and its inverse in an analytic or closed form, so that relation 7.22 is a known vector function to be evaluated at each step.

2. If the initial value problem 7.14 does not lead to a closed form solution for all values of $\underline{\zeta}$ or if that solution is too complex to be useful in forming relation 7.22, then Newton's method cannot be carried out exactly (or at least not conveniently so).

1. Approximate Newton's Method

In the second case above, an iteration scheme which does not require the derivative should be used. There are many such methods possible and a considerable literature exists. For a simple treatment the books by Froberg,[31] Henrici,[40] and Fox[30] might be mentioned. The book by Kantorovich and Akilov,[49] and that edited by Todd[3] provide more advanced and additional material.

One of the schemes available utilizes a "moving secant" as a replacement for the derivative.

For an ordinary function in one dimension,

$$T(x) = 0$$

the recursive relation is

$$x^{i+1} = x^i - \left[\frac{T(x^i) - T(x^{i-1})}{x^i - x^{i-1}}\right]^{-1} \cdot T(x^i) \qquad (7.23)$$

A similar method using a "fixed secant" should be mentioned. Its recursive relation is,

$$x^{i+1} = x^i - \left[\frac{T(x^1) - T(x^0)}{x^1 - x^0}\right]^{-1} \cdot T(x^i) \qquad (7.24)$$

The fixed secant method is simpler, but does not generally converge as rapidly as the moving secant method. Neither of them has an asymptotic convergence as rapid as that of Newton's method.

These methods can be adapted for the vector case. Let the $i^{\underline{th}}$ element of the vector operator $T(\underline{\pi})$ be denoted by $T(\underline{\pi})_i$. As noted in Chapter II, the derivative of the operator $T(\underline{\pi})$ is a matrix, the $ij^{\underline{th}}$ element of which is

$$\frac{\partial T(\underline{\pi})_i}{\partial \pi_j}$$

To estimate this matrix by secants requires (at least) $n+1$ evaluations of the operator T at different values of $\underline{\pi}$. Then the matrix must be inverted. Because of this lengthy procedure, the larger n is the more attractive the fixed secant method becomes as compared with the moving secant method. In practice, the two can be mixed, i.e.,

> Start with one step of the moving secant method, then add several steps using the fixed secant. Now reevaluate the secant matrix for another step of the moving secant method, then add several steps using the new fixed secant, etc.

One method of finding the secant matrix is as follows:

> Let the present estimate of the solution be a vector $\underline{\pi}_0$. Form n other vectors $\underline{\pi}_j$ by adding a small quantity δ to the $j^{\underline{th}}$ component of $\underline{\pi}_0$. Evaluate the $n+1$ operators
>
> $$T(\underline{\pi}_j) \qquad j = 0, 1, \ldots n$$
>
> Then the desired secant matrix is

$$\frac{\Delta T(\underline{\pi})}{\Delta \underline{\pi}} = \left[\frac{1}{\delta}\ T(\underline{\pi}_1) \vdots T(\underline{\pi}_0), \ldots, T(\underline{\pi}_n) \vdots\ T(\underline{\pi}_0)\right] \qquad (7.25)$$

Thus the moving secant method is

$$\underline{\pi}^{i+1} = \underline{\pi}^{i} - \left[\frac{\Delta T(\underline{\pi}^{i})}{\Delta \underline{\pi}^{i}}\right]^{-1} \cdot T(\underline{\pi}^{i}) \qquad (7.26)$$

As a final note, it should be mentioned that there is a method called Steffensen's iteration, outlined in Henrici,[40] Chapter 5, Section 9. Henrici states that "Substantial experimental evidence, and also some theoretical considerations, seem to indicate, however, that the algorithm is indeed quadratically convergent in a large number of cases, even when ordinary iteration diverges." The good convergence properties noted should make this an attractive alternate to the secant method outlined above.

F. A SIMPLER PROBLEM

Problem 2 as stated is very difficult to work with. By suitably restricting the form of the plant equation 7.1 and the cost functional 7.3 one can guarantee an analytic expression for the operator 7.19. Then the recursive relation for Newton's method also leads to an analytic expression. This avoids any necessity of resorting to the approximate methods mentioned in Section E, although the analytic expression can be so complex that the approximate method is easier to handle.

Suppose the Hamiltonian 7.4 is linear in the state variables and has no state variable-control variable cross-products. Then the costate half 7.10 of the two-point boundary value problem is independent of the state. One way to insure this is as follows:

Problem 3

Similar to Problem 2 except that the state equations have the form

$$\dot{\underline{x}}(t) = \underline{A}(t)\underline{x}(t) + \underline{f}(\underline{u}(t), t) \qquad (7.27)$$

and the cost functional has the form

$$J(\underline{u}) = \int_{t_0}^{t_1} L(\underline{u}(\tau), \tau)\, d\tau \qquad (7.28)$$

Now let us find the operator $T(\underline{\pi})$ for this restricted problem. First of all the Hamiltonian is,

$$H(\underline{x},\underline{u},\underline{p},t) = L(\underline{u},t) + \underline{p}'(t)\,\underline{A}(t)\underline{x}(t) + \underline{p}'(t)\underline{f}(\underline{u},t) \tag{7.29}$$

and the costate is governed by

$$\dot{\underline{p}}(t) = -\underline{A}'(t)\,\underline{p}(t) \tag{7.30}$$

The main point of this restricted problem is that the costate equations can be integrated separately. Since they are linear, there is a fundamental matrix $\underline{\psi}(t,t_0)$ such that

$$\underline{p}(t) = \underline{\psi}(t,t_0)\underline{\pi} \tag{7.31}$$

for any initial condition vector $\underline{\pi}$. Consider the fundamental matrix $\underline{\phi}(t,t_0)$ of the linear part of the state equations. That is

$$\dot{\underline{x}}(t) = \underline{A}(t)\underline{x}(t)$$

implies $$\underline{x}(t) = \underline{\phi}(t,t_0)\underline{\xi}$$

The costate equations are the adjoint to this, so that (see for example Athans and Falb,[4] page 147).

$$\underline{\psi}(t,t_0) = \underline{\phi}'(t_0,t) \tag{7.32}$$

Substituting Eqs. 7.32 into 7.31 yields

$$\underline{p}(t) = \underline{\phi}'(t_0,t)\underline{\pi} \tag{7.33}$$

The two-point boundary value problem proceeds as before, except that the form of the Hamiltonian 7.29 leads to an optimal control relation $\underline{u}^*$ which is not dependent on the state of the system.

$$\underline{u}^*(t) = \underline{v}[\underline{p}^*,t]$$

or with Eq. 7.33

$$\underline{u}^*(t) = \underline{v}[\underline{\phi}'(t_0,t)\underline{\pi}^*,t] \tag{7.34}$$

Substituting Eq. 7.34 into the state equations yields,

$$\dot{\underline{x}}(t) = \underline{A}(t)\underline{x}(t) + \underline{f}(\underline{v}[\underline{\phi}'(t_0,t)\underline{\pi}^*,t],t) \tag{7.35}$$

In a formal way, the composite nonlinearity g is defined such that

$$\underline{g}[\cdot,t] = \underline{f}(\underline{v}[\cdot,t],t)$$

It is assumed that:

$$\underline{f}: R_d \times [t_0,t_1] \rightarrow R_n$$

and

$$\underline{v}: R_n \times [t_0,t_1] \rightarrow R_d$$

except possibly for a set of measure zero, so that the operation can almost always be carried out. Then a composite function does exist, except possibly on a set of measure zero.

Finally, the state equations, 7.35, have the integral form

$$\underline{x}(t) = \underline{\Phi}(t,t_0)\,\underline{\xi} + \int_{t_0}^{t} \underline{\Phi}(t,\tau)\,\underline{g}[\underline{\Phi}'(t_0,\tau)\,\underline{\pi},\tau]\,d\tau \qquad (7.36)$$

so that the operator $T(\underline{\pi})$ becomes,

$$T(\underline{\pi}) = \underline{\Phi}(t_1,t_0)\,\underline{\xi} + \int_{t_0}^{t_1} \underline{\Phi}(t_1,\tau)\,\underline{g}[\underline{\Phi}'(t_0,\tau)\,\underline{\pi},\tau]\,d\tau - \underline{\theta} \qquad (7.37)$$

and the recursive relation of Newton's method can be written down explicitly. As in Chapter II, the first derivative is a matrix

$$\underline{\pi}^{i+1} = \underline{\pi}^{i} - \left[\int_{t_0}^{t_1} \underline{\Phi}(t_1,\tau)\,\underline{\Phi}'(t_0,\tau)\,\underline{g}^{(1)}[\underline{\Phi}'(t_0,\tau)\,\underline{\pi}^{i},\tau]\,d\tau\right]^{-1} \{\underline{\Phi}(t_1,t_0)\,\underline{\xi} - \underline{\theta}$$

$$+ \int_{t_0}^{t_1} \underline{\Phi}(t_1,\tau)\,\underline{g}[\underline{\Phi}'(t_0,\tau)\,\underline{\pi}^{i},\tau]\,d\tau\} \qquad (7.38)$$

In this case $\underline{g}^{(1)}[\cdot,t]$ means the derivative with respect to the first argument, holding the time fixed.

Comment 7:2 An alternate formulation has been published by Witsenhausen,[97] pages 9-10. In this application the state and costate equations would be combined into one set of 2n equations $\underline{y}(t)$. Assume these equations have some linear terms, so that

$$\dot{\underline{y}}(t) = \underline{A}(t)\,\underline{y}(t) + \underline{f}(\underline{y}(t), t) \tag{7.39}$$

The boundary conditions are given in a more general form

$$\underline{N}\,\underline{y}(t_0) + \underline{M}\,\underline{y}(t_1) = \underline{c} \tag{7.40}$$

Use Eq. 7.40 to eliminate $\underline{y}(t_0)$ from the fundamental solution of Eq. 7.39. The result is a (vector) Fredholm integral equation

$$\underline{y}(t) = \underline{G}_1(t)\,\underline{c} + \int_{t_0}^{t_1} \underline{G}(t,\tau)\,\underline{f}(\underline{y}(\tau),\tau)\,d\tau \tag{7.41}$$

where

$$\underline{G}_1(t) = \underline{\phi}(t,t_0)\left[\,\underline{N} + \underline{M}\,\underline{\phi}(t_1,t_0)\right]^{-1}$$

$$\underline{G}(t,\tau) = \begin{cases} \underline{G}_1(t)\,\underline{N}\,\underline{\phi}(t_0,\tau) & \text{for } \tau < t \\ -\underline{G}_1(t)\,\underline{M}\,\underline{\phi}(t_1,\tau) & \text{for } \tau > t \end{cases}$$

Assuming, of course, that the matrix $[\,\underline{N} + \underline{M}\,\underline{\phi}(t_1,t_0)]$ has an inverse. This formulation is more general than the one used in this thesis.

In summary, it is possible to treat the much more general Problem 2 by an approach similar to that used in Chapter II for Problem 1. However, one cannot handle the relations with the same assurance. There may be more than one extremal solution; the operator $T(\underline{\pi})$ may be so complex that Newton's method is difficult to handle; an analytic expression may not be available for the form of the optimal control $\underline{u}^*(t)$; and so on. One conclusion is that there is a trade-off between the complexity of the problem handled and the ease of carrying out this approach.

CHAPTER VIII

POSSIBLE EXTENSIONS

There are a number of ways in which the procedure and the computer program can be changed and extended to accommodate different problems. Some of these are described in this chapter. The changes described are not mutually exclusive; that is, several of them might occur in the same problem. With some of these extensions the convergence and uniqueness properties shown in Chapter III are no longer guaranteed.

A. SEVERAL CONTROL VARIABLES

One of the easiest extensions to make would be to change to a vector control variable. The scalar u becomes a vector $\underline{u}$ and the vector $\underline{b}$ becomes a matrix $\underline{B}$. All the equations of Chapter II carry through with this change. For instance the state equations are,

$$\dot{\underline{x}}(t) = \underline{A}\,\underline{x}(t) + \underline{B}\,\underline{u}(t)$$

and the control components are

$$u_i^*(t) = -\,\mathrm{dez}\,(\underline{b}_i'\,\underline{e}^{-\underline{A}'t}\underline{\pi}) \tag{8.1}$$

where $\underline{b}_i$ is the $i^{\underline{th}}$ column of the matrix $\underline{B}$. It is assumed that each component of the vector $\underline{u}$ is constrained to lie in the interval $[-1, +1]$. A better way to formulate this is to define the vector deadzone function, whose $i^{\underline{th}}$ component is the deadzone function of its $i^{\underline{th}}$ argument.

$$\underline{u}^*(t) = -\underline{DEZ}\,(\underline{B}'e^{-A't}\underline{\pi}) \tag{8.2}$$

From the computational point of view, the main change is that instead of having a vector function $\underline{q}(t)$ to store, there is a matrix function $\underline{Q}(t)$ to either store or else compute at each iteration.

$$\underline{Q}(t) = \underline{e}^{-\underline{A}t}\underline{B} \tag{8.3}$$

The costate initial condition $\underline{\pi}$ is still a vector, and so many of the graphs will be similar to those from Chapter V.

B. TIME VARYING EQUATIONS

If the equations of state are time varying but still linear the matrix exponential is replaced by the more general fundamental matrix $\underline{\Phi}(t, t_0)$. In this case the time interval will be indicated by $[t_0, t_1]$. Once the fundamental matrix is computed and the vector function $\underline{q}(t)$ stored the computation proceeds as shown in Chapter II. One straightforward way to compute $\underline{\Phi}^{-1}(t, t_0)$ is directly from the differential equation.

$$\underline{\Phi}'(t_0, t_0) = \underline{I} \tag{8.4}$$

and

$$\dot{\underline{\Phi}}'(t_0, t) = -\underline{A}'(t)\underline{\Phi}'(t_0, t) \tag{8.5}$$

After the matrix $\underline{\Phi}(t_0, t) = \underline{\Phi}^{-1}(t, t_0)$ is computed for each time step t_i, the vector $\underline{q}(t_i)$ is stored.

$$\underline{q}'(t_i) = \underline{b}'\underline{\Phi}'(t_0, t_i) \tag{8.6}$$

The matrix $\underline{\Phi}'(t_0, t_i)$ does not need to be saved. However at the final step the matrix $\underline{\Phi}'(t_0, t_1)$ should be transposed and saved for use where $\underline{e}^{-\underline{A}T}$ was used in the original program.

C. DIFFERENT COST CRITERIA

Suppose the cost remains a functional only of the control variable, plus perhaps the time variable and/or a linear combination of the state variables. For example, let

$$J(u) = \int_{t_0}^{t_1} g(u(\sigma), \sigma)d\sigma \tag{8.7}$$

then the Hamiltonian becomes,

$$H(\underline{x}, u, \underline{p}, t) = g(u(t), t) + \underline{p}'(t)\underline{A}\,\underline{x}(t) + \underline{p}'(t)\underline{b}\,u(t)$$

The expressions for the costate and for the optimal control do not contain the state variables. If the expression for the control can be solved explicitly for the control variable, one has the equation below.

$$u^*(t) = h(\underline{p}(t), t) \tag{8.8}$$

Since the Hamiltonian and the control variable are scalar functions, Eq. 8.8 involves solving a scalar algebraic equation. In most examples of interest this function will have some finite number of discontinuities. As is well known from the theory of approximations it can then be approximated as closely as desired, in the L_2 norm on any bounded subset in the space of its arguments $R_n x[t_0, t_1]$, by a smooth function (having derivatives of all orders).

With this in mind, a sequence of approximations should be designed which converges to the optimal control relation 8.8, similar to the sequence $\{u_k(t)\}$ in Chapter II. The exponential form used there is convenient, but see Section D of this chapter for some other possible approximations.

Once the approximate control functions have been fixed, the computations proceed as in Chapter II.

D. CHOICE OF APPROXIMATE CONTROL FUNCTIONS u_k

The exponential form of approximate control function chosen in Chapter II is convenient, but there are many other possible ways of forming the approximation. A few possibilities are described below.

1. Distribution functions would provide a possible way of forming the approximation. These "generalized functions" were developed by Schwartz,[88] and are presented by Zadeh and Desoer[98] in Appendix A, and by Beckenbach[14] in Chapter I (by Erdelyi). Their chief advantage lies in the theoretical framework which insures that any distribution function possesses derivatives of all orders. Since the optimal control function $u^*(\cdot)$ is the sum of two step functions, its derivatives do not exist in the ordinary sense, but are sometimes represented symbolically by delta functions $\delta(t)$ and their derivatives $\delta^{(n)}(t)$. By definition the delta function is the derivative, in the distribution sense, of the unit step function.

$$\delta(t) = \frac{d}{dt} 1(t)$$

so that
$$\frac{d}{dt} \mathrm{dez}(t) = \delta(t-1) + \delta(t+1) \tag{8.9}$$

and
$$\frac{d^2}{dt^2} \mathrm{dez}(t) = \delta^{(1)}(t-1) + \delta^{(1)}(t+1)$$

etc.

A typical approximate function and its first derivative are shown in Fig. 8.1. Notice that the approximate function u_k is identical

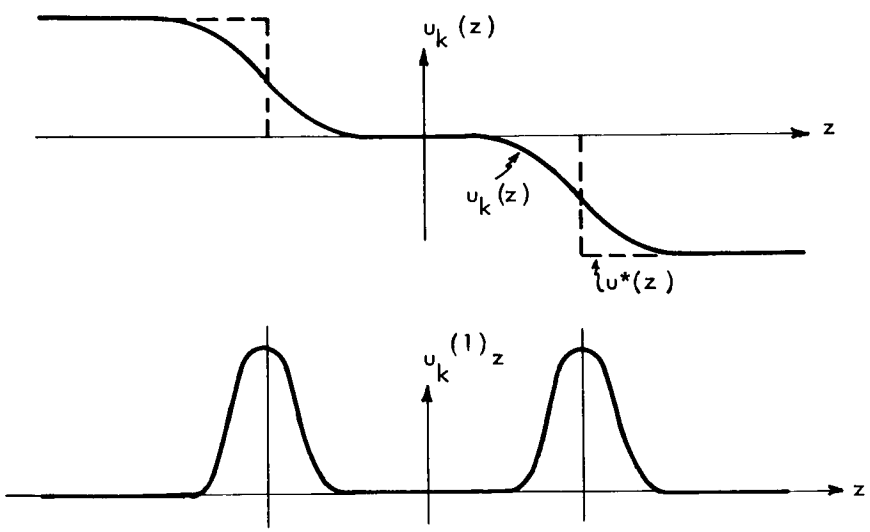


Fig. 8.1 Distribution Function Approximation to the Optical Control

with the optimal function u^* except in the intervals near its discontinuities.

If a sequence of approximations to the optimal control function $u^*(\cdot)$ is formed by using distribution functions, the derivatives are guaranteed to exist for each member of the sequence as well as for its limit. Thus a theoretical justification is available for the application of Newton's method to the exact operator $T(\underline{\pi})$. In addition the convergence theorem of Kantorovich can be examined using the exact operator, although the approximation used in Chapter VII in bounding the norm of the second derivative operator would yield an infinite value for the convergence parameter h.

2. A polynomial could be used for the approximate function. Much is known about the properties and use of polynomials as approximating functions. Since the function to be approximated is odd (not even) only polynomials of odd order would be used. As before the sequence would start with a straight line approximation--a first order polynomial.

One way to proceed would be to increase the order of the polynomial by two with each new member of the sequence. The coefficients must be chosen so that the sequence converges toward the optimal control function $u^*(\cdot)$. If any operator has to be redefined, the coefficients of that polynomial are adjusted to reduce the distance between it and the previous polynomial. In this case it might be advisable to use two or more polynomials of the same order. Otherwise the general polynomial $u_k(x)$ would be,

$$u_k(x) = a_k x^{2k+1} + a_{k-1} x^{2k-1} + \ldots + a_0 x \tag{8.10}$$

3. Several stages can be used in the sequence of approximate control functions $\{u_k\}$. For example: Start as before with a linear function.

$$u_0(z) = a\,z$$

Break the abcissa into an outside part for $|z| > 1$ and an inside part for $|z| < 1$. Now using three straight line segments, bring the slope to 1.0 in the inside segment and reduce it to 0.0 in both outside segments as shown in Fig. 8.2. This may be done in several steps if

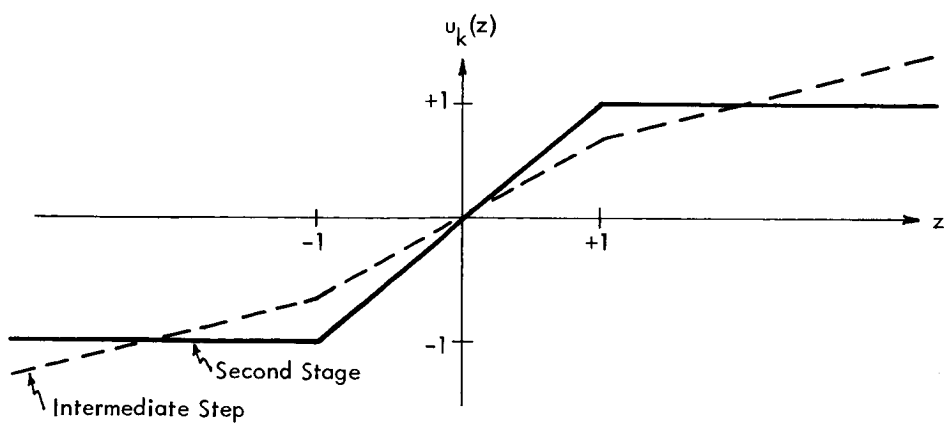


Fig. 8.2 Straight Line Approximate Control Function

it requires more than one member of the sequence to assure sequential convergence. Finally, reduce the slope of the "inside" segment to 0, producing discontinuities at $z = \pm 1$.

The chief disadvantages are that discontinuities must be handled in using Newton's method, and the convergence theorem of Kantorovich gives infinity with the bound used on the second derivative operator. The advantage is the greater ease of computation with straight line segments. With the linear time-invariant plant the integration can be done analytically for each segment.

4. A family of empirical curves can be used for the sequence of approximate controls. Suppose a control system is being designed to approximate the action of the fuel optimal control. Clearly by spending more money to make the control device larger or more intricate the fuel optimal control can be approached more closely. The procedure is to take the expected characteristics of several of these devices, of increasing cost, as the sequence of approximate controls. Now the computer results give the overall system characteristics with the sequence of devices, and permit a trade-off study between control device cost and system performance.

E. NONLINEAR EQUATIONS

The procedure of this thesis can be carried out with nonlinear differential equations of a rather general nature. However the length of the calculations involved may become prohibitive. A general approach is outlined in Chapter VII. As indicated, a great deal of flexibility exists in treating the nonlinearities. In Section F of Chapter VII a simpler problem called Problem 3 is treated in which the nonlinearities do not involve the state variables, and the computation is not so difficult in this case.

F. OTHER TERMINAL BOUNDARY CONDITIONS

1. In the cases studied, all the terminal conditions were fixed, both the terminal time t_1 and the final state vector θ. If any of the state variables are not fixed at the given final time, then the corresponding costate variable is fixed at time $t = t_1$ as a consequence

of Pontryagin's maximum principle.[84] This reduces the number of variables to be found. For example, in Problem 1, let the first r components of the state vector be fixed at time t = T, and define the r vector $\underline{\theta}_j$.

$$\underline{\theta}_j = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_r \end{bmatrix}$$

Define the costate <u>final</u> condition vector $\underline{\pi}_f$ and partition it.

$$\underline{\pi}_f = \begin{bmatrix} \underline{\pi}_j \\ --- \\ \underline{\pi}_\ell \end{bmatrix}$$

where $\underline{\pi}_\ell$ is now the known final boundary condition on the costate, given by comment 8.1 below. Also, the fundamental matrix must be partitioned.

$$e^{\underline{A}t} = \begin{bmatrix} \underline{e}_{jj} & \underline{e}_{j\ell} \\ --- & --- \\ \underline{e}_{\ell j} & \underline{e}_{\ell\ell} \end{bmatrix}$$

Note: In each of these definitions it is understood that

$$1 \leq j \leq r \quad \text{and} \quad r+1 \leq \ell \leq n$$

The vector $\underline{q}(t)$ is changed to account for the <u>final</u> costate vector $\underline{\pi}_f$. Let

$$\underline{q}_f(t) = e^{-\underline{A}(t-T)}\,\underline{b} = e^{\underline{A}T}\,\underline{q}(t) \tag{8.11}$$

and

$$\underline{q}_f(t) = \begin{bmatrix} \underline{q}_j(t) \\ ---- \\ \underline{q}_\ell(t) \end{bmatrix}$$

Then the optimal control function is given by

$$u(t) = -\text{dez}[\underline{q}_f'(t)\underline{\pi}_f] = -\text{dez}[\underline{q}_j'(t)\underline{\pi}_j + \underline{q}_\ell'(t)\underline{\pi}_\ell] \tag{8.12}$$

The corresponding state trajectory is

$$\underline{x}(t) = \underline{e}^{\underline{A}t}[\underline{\xi} - \int_0^t \underline{q}(\tau)\,\mathrm{dez}[\underline{q}_f'(\tau)\,\underline{\pi}_f]\,d\tau] \tag{8.13}$$

and the operator $T(\underline{\pi}_j)$ is now defined on the space R_r.

$$T(\underline{\pi}_j) = -\underline{\theta}_j + [\underline{e}_{jj} \vdots \underline{e}_{j\ell}][\underline{\xi} - \int_0^T \underline{q}(\tau)\,\mathrm{dez}[\underline{q}_j'(\tau)\underline{\pi}_j + \underline{q}_\ell'(\tau)\underline{\pi}_\ell] \cdot d\tau] \tag{8.14}$$

Just as in Chapter II, the deadzone function can be replaced by an approximate function u_k and the corresponding approximate operator $T_k(\underline{\pi}_j)$.

$$T_k(\underline{\pi}_j) = -\underline{\theta}_j + [\underline{e}_{jj} \vdots \underline{e}_{j\ell}][\underline{\xi} - \alpha_k \underline{W}_f(T)\underline{\pi}_f - \int_0^T \underline{q}(\tau)u_k[\underline{q}_f'(\tau)\underline{\pi}_f]\,d\tau] \tag{8.15}$$

and $$\underline{W}_f(T) = \int_0^T \underline{q}(\tau)\underline{q}_f'(\tau)\,d\tau$$

The first derivative operator is an r x r matrix.

$$T_k^{(1)}(\underline{\pi}_j) = -[\underline{e}_{jj} \vdots \underline{e}_{j\ell}][\alpha_k \underline{W}_f(T) + \int_0^T \underline{q}(\tau)\underline{q}_j'(\tau)u_k^{(1)}[\underline{q}_j'(\tau)\underline{\pi}_j + \underline{q}_\ell'(\tau)\underline{\pi}_\ell]\,d\tau \tag{8.16}$$

Finally, the recursion relation of Newton's method is

$$\underline{\pi}_j^{i+1} = \underline{\pi}_j^i - [T_k^{(1)}(\underline{\pi}_j^i)]^{-1}\, T_k(\underline{\pi}_j^i) \tag{8.17}$$

Equation 8.17 can be written out fully by substituting Eqs. 8.15 and 8.16 into it.

Comment 8.1: The costate boundary condition on $\underline{\pi}_\ell$ is given in general form by (see e.g., Athans and Falb,[4] p. 306).

$$\underline{\pi}_\ell = \frac{\partial J(\underline{x}, u, t)}{\partial \underline{x}_\ell(T)}$$

where

$$\underline{x}_\ell = \begin{bmatrix} x_{r+1} \\ \vdots \\ x_n \end{bmatrix}$$

Using the penalty function of Chapter II, which does not depend on the final state vector $\underline{x}_\ell(T)$, yields

$$\underline{\pi}_\ell = \odot$$

If none of the state variables are fixed at the terminal time t_1, then a complete set of costate final conditions is available. Under these conditions Problem 1 or Problem 3 (of Chapter VII) has a closed form solution. Let the final condition on the costate be

$$\underline{p}(t_1) = \underline{\pi}_f$$

Then the costate is

$$\underline{p}(t) = \underline{e}^{-\underline{A}'(t-t_1)} \underline{\pi}_f$$

For Problem 1, the optimal control is,

$$u^*(t) = -\text{dez}\left[\underline{q}'(t)\underline{e}^{\underline{A}'T}\underline{\pi}_f\right] = -\text{dez}\left[\underline{q}_f'(t)\underline{\pi}_f\right] \tag{8.18}$$

and the state is,

$$\underline{x}(t) = \underline{e}^{\underline{A}t}\left[\underline{\xi} - \int_0^t \underline{q}(\tau)\,\text{dez}\left[\underline{q}'(\tau)\underline{e}^{\underline{A}'T}\underline{\pi}_f\right] d\tau\right] \tag{8.19}$$

For Problem 3, the optimal control is,

$$\underline{u}^*(t) = v\left[\underline{\phi}'(t, t_1)\underline{\pi}_f,\ t\right] \tag{8.20}$$

and the state is,

$$\underline{x}(t) = \underline{\Phi}(t, t_0)\underline{\xi} + \int_{t_0}^{t} \underline{\Phi}(t, \tau)\underline{g}[\underline{\Phi}'(\tau, t_1)\underline{\pi}_f, \tau]\, d\tau \tag{8.21}$$

2. If the final time t_1 (or T) is not specified but is to be finite, a change must be made in the procedure, since the integrals appearing in the recursive scheme require fixed limits. There are various ways of adapting the problem. One way is to choose a different independent variable. If one of the state variables $x_i(t)$ is fixed at both $t = t_0$ and $t = t_1$, and varies monotonically in between, that variable can be chosen as the independent variable, and used as the basis of the integrations.

Another approach is to treat the terminal time t_1 as an extra variable in the interations of Newton's method. A rather general development of this idea is given by Kelley in Chapter 6, Section 26 of Leitmann,[53] in connection with the use of a gradient method. Normally, there will be a stopping criterion,

$$s(\underline{x}(t_1), t_1) = 0 \tag{8.22}$$

A straightforward application of Newton's method yields the recursion relation for t_1.

$$t_1^{i+1} = t_1^i - \left[\frac{\partial s(\underline{x}(t_1^i), t_1^i)}{\partial t_1^i}\right]^{-1} \cdot s(\underline{x}(t_1^i), t_1^i) \tag{8.23}$$

If the stopping condition does not contain the terminal time t_1 explicitly or has a weak dependence on t_1, a total derivative can be used. Let

$$\frac{Ds}{Dt_1} = \frac{\partial s}{\partial t_1} + \left[\frac{\partial s}{\partial \underline{x}}\right]' \frac{d\underline{x}}{dt_1}$$

$$= \frac{\partial s(\underline{x}(t_1), t_1)'}{\partial \underline{x}(t_1)} \cdot \underline{f}(\underline{x}(t_1), \underline{u}(t_1), t_1) + \frac{\partial s(\underline{x}(t_1), t_1)}{\partial t_1}$$

where $\underline{\dot{x}} = \underline{f}(\underline{x}, \underline{u}, t)$ is the system differential equation 7.1. Then Eq. 8.23 is replaced by,

$$t_1^{i+1} = t_1^i - \left[\frac{\partial s(\underline{x}(t_1^i), t_1^i)'}{\partial \underline{x}(t_1^i)} \cdot \underline{f}(\underline{x}(t_1^i), \underline{u}(t_1^i), t_1^i) + \frac{\partial s(\underline{x}(t_1^i), t_1^i)}{\partial t_1^i} \right]^{-1} \cdot$$

$$\cdot s(\underline{x}(t_1^i), t_1^i) \tag{8.24}$$

As an example, consider the problem of time optimal control to a fixed point (usually the origin). The most natural stopping condition is the terminal value of the Hamiltonian.

$$H(\underline{x}(t_1), \underline{u}(t_1), \underline{p}(t_1), t_1) = 0 \tag{8.25}$$

In this case, the stopping condition also contains the control variable $\underline{u}(t_1)$. However in satisfying the second necessary condition of the Minimum Principle (minimization of the Hamiltonian) we normally require

$$\frac{\partial H(\underline{x}, \underline{u}, \underline{p}, t)}{\partial \underline{u}(t)} = 0$$

In addition, the necessary conditions lead to,

$$\frac{\partial H(\underline{x}, \underline{u}, \underline{p}, t)'}{\partial \underline{x}(t)} \cdot \frac{d\underline{x}(t)}{dt} = 0$$

For simplicity, define

$$H(t_1) \equiv H(\underline{x}(t_1), \underline{u}(t_1), \underline{p}(t_1), t_1) \tag{8.26}$$

Then the recursive relation 8.24 becomes,

$$t_1^{i+1} = t_1^i - \frac{H(t_1^i)}{\dfrac{\partial H(t_1^i)}{\partial t_1^i}} \tag{8.27}$$

As an alternate stopping condition, consider the function

$$s(\underline{x}(t_1)) = \frac{1}{2}\underline{x}'(t_1)\underline{x}(t_1) = 0 \tag{8.28}$$

Then the recursive relation 8.24 becomes,

$$t_1^{i+1} = t_1^i - \frac{\frac{1}{2}\underline{x}'(t_1^i)\underline{x}(t_1^i)}{\underline{x}'(t_1^i)\underline{f}(\underline{x}(t_1^i), \underline{u}(t_1^i), t_1^i)} \tag{8.29}$$

Finally, if the stopping condition is taken to be

$$s(\underline{x}(t_1)) = \sum_{j=1}^{n} x_j(t_1) = 0$$

then the recursive relation becomes,

$$t_1^{i+1} = t_1^i - \frac{\sum_{j=1}^{n} x_j(t_1)}{\sum_{j=1}^{n} f_j(\underline{x}(t_1^i), \underline{u}(t_1^i), t_1^i)}$$

An initial guess is required for the terminal time t_1. Then the rest of the problem formulation proceeds as outlined in Section 8.3 (and also Section 8.5 if the nonlinear plant is used).

Making changes in the procedure, or altering the problem itself to ease the computational problem is really an art. There are other ways in which the iteration for the terminal time could be handled, and also many different ways of expressing the stopping function $s(\underline{x}(t_1), t_1)$. There are also different and more general ways of formulating the optimal control problem which could be considered, but these are outside the range of this thesis.

APPENDIX A

NOTATION AND BASIC CONCEPTS

The purpose of this appendix is first to establish the notation and nomenclature* of the thesis, and second to define certain control theory notions and theorems. This is only a handy reference for concepts needed in the main text, and is neither complete nor rigorous. A more comprehensive development is found for instance in Athans and Falb.[4]

1. NOTATION

Vector notation is used extensively. Some set theory concepts are also used. As far as possible, the notation is similar to that in recent control and systems books such as Athans and Falb[4] or Zadeh and Desoer.[98] Theorems, comments, etc. are numbered consecutively within each chapter. When referred to outside the chapter, the chapter number is included. Thus Theorem 3.1 is the first Theorem in Chapter III.

Column Vectors are indicated by underlined lower case letters, and matrices by underlined upper case letters. The transpose of a vector or matrix is indicated by an apostrophe after the letter. Elements of a vector or matrix are indicated by single or double subscripts on lower case letters.

For example, in two dimensions;

$$\underline{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

$$\underline{B} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

$$\underline{y}' = \begin{bmatrix} y_1 & y_2 \end{bmatrix}$$

*A list of the nomenclature used is found at the end of the appendices.

$$\underline{B}' = \begin{bmatrix} b_{11} & b_{21} \\ b_{12} & b_{22} \end{bmatrix}$$

If a square matrix $\underline{B}$ of order n has nonzero determinant (is of rank n), its inverse is denoted by $\underline{B}^{-1}$. Thus, given the linear relations,

$$\underline{B}\,\underline{y} = \underline{z} \tag{A.1}$$

and the condition that

$$\det \underline{B} \neq 0 \tag{A.2}$$

one can solve for the vector $\underline{y}$ in Eq. A.1.

$$\underline{y} = \underline{B}^{-1}\,\underline{z} \tag{A.3.}$$

The exponential matrix of a matrix $\underline{B}$ is defined to be

$$\underline{e}^{B} \equiv \sum_{j=0}^{\infty} \frac{1}{j!} (\underline{B})^{j}$$

so also

$$\underline{e}^{\underline{B}t} = \sum_{j=0}^{\infty} \frac{1}{j!} (Bt)^{j} \tag{A.4}$$

The time derivative of a function is indicated by a dot placed over it. Thus

$$\frac{dw(t_1)}{dt} \equiv \dot{w}(t_1) \tag{A.5}$$

indicates the time derivative of the function $w(t)$ evaluated at time t_1.

The partial derivative of a function of one (possibly vector valued) argument with respect to that argument is indicated by a superscript (1). Thus

$$\frac{\partial f(\cdot)}{\partial \cdot} \equiv f^{(1)}(\cdot)$$

e.g.,

$$\frac{\partial u(\underline{q}'(t)\underline{\pi})}{\partial(\underline{q}'(t)\underline{\pi})} = u^{(1)}(\underline{q}(t)\underline{\pi}) \tag{A.6}$$

When used with an operator in function space this is defined to mean the Frechet derivative, as defined in Appendix B.

Notice that the tensor rank of the partial derivative depends on the nature of the argument. For example, given a vector valued function of a vector valued argument, say $\underline{f}(\underline{y})$, then $\underline{f}^{(1)}(\underline{y})$ is a matrix and $\underline{f}^{(2)}(\underline{y})$ is a third order tensor.

2. SETS

A <u>set</u> is a collection of mathematical objects related to each other in some way. Upper case letters are used to denote sets. The statement "s belongs to set S" is written

$$s \in S. \tag{A.7}$$

If the set S is composed of those elements s which have some property S, the following notation is used.

$$S = \{s: s \text{ has property } S\} \tag{A.8}$$

For example if S consists of all real numbers of magnitude less than 1,

$$S = \{s: |s| < 1\}$$

If S_1 is some other set, all of whose members are contained in S, then

$$S_1 \subset S$$

The statement $S_1 = S_2$ implies the two statements $S_1 \subset S_2$ and $S_2 \subset S_1$. The empty set having no members is denoted by the symbol $\odot$. The <u>direct product</u> of two sets S_1 and S_2, written $S_1 \times S_2$ is

the set of all pairs of elements (s_1, s_2) such that $s_1 \in S_1$ and $s_2 \in S_2$.

$$S_1 \times S_2 = \{(s_1, s_2) : s_1 \in S_1 \text{ and } s_2 \in S_2\}$$

A linear set is one for which the operations of addition and multiplication by a scalar are defined and satisfy the following conditions for arbitrary members s_1, s_2, and s_3 of a set S.

0) $(s_1+s_2) \epsilon S$ and $c_1 s_1 \epsilon S$, for all s_1, s_2 and all real numbers c_1 and c_2.

1) $(s_1+s_2) + s_3 = s_1 + (s_2+s_3)$

2) $s_1+s_2 = s_2+s_1$

3) A null element $\odot$ exists in S such that $o \cdot s = \odot$ for all $s \in S$. (A.9.)

4) $(c_1+c_2)s = c_1 s + c_2 s$

5) $c(s_1+s_2) = cs_1+cs_2$

6) $(c_1 c_2)s = c_1(c_2 s)$

7) $1 \cdot s = s$

The set of most use is the n dimensional vector space R_n, each element of which is an ordered n-tuple of real numbers. This is a linear space. An element of the set is written as an n dimensional vector, e.g., as

$$\underline{s} = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{bmatrix}$$

The elements of $\underline{s}$ are called linearly independent if no one of them can be made equal to a linear combination of the others.

The real line is itself a space (i.e., the space R_1). A closed interval on the real line will be indicated by brackets, and an open interval by parentheses. Mixtures are permitted, e.g.,

$$(0, T] = \{t: 0 < t \leq T\} \tag{A.10}$$

Another important space is the space C of all possible continuous functions defined on a closed interval $[t_0, t_1]$ of the real line. A space consisting of n such functions will be called the space C_n. The space of functions with m continuous derivatives is called the space $C^{(m)}$. For more information see any basic book on analysis, such as Diendonné.[25]

3. NORMS

A concept of obvious importance in the study of convergence of numerical methods is that of the distance between any two members s_1 and s_2 of a set S written $\rho(s_1, s_2)$. There are many possible ways of defining distance, but the most useful ones satisfy the metric space axioms below

$$\begin{aligned} &\text{i} \quad \rho(s_1, s_2) \geq 0 \text{ and } \rho(s_1, s_2) = 0 \text{ implies } s_1 = s_2 \\ &\text{ii} \quad \rho(s_1, s_2) = \rho(s_2, s_1) \\ &\text{iii} \quad \rho(s_1, s_2) \leq \rho(s_1, s_3) + \rho(s_3, s_2) \end{aligned} \tag{A.11}$$

Once a distance function is chosen the set S becomes a metric space.

If the set happens to be a linear set one can take advantage of this by defining a more restrictive kind of distance function called a norm written $\|s\|$ which satisfies the normed space axioms given below:

$$\begin{aligned} &\text{i} \quad \|s\| = 0 \text{ is equivalent to } s = \odot \\ &\text{ii} \quad \|cs\| = |c| \cdot \|s\| \\ &\text{iii} \quad \|s_1 + s_2\| \leq \|s_1\| + \|s_2\| \end{aligned} \tag{A.12}$$

Since the space R_n, of primary use in this thesis, is a linear space, norms are used as a measure of distance or of size.

An important type of norm in a finite dimensional space is called the ℓ_p norm. It is defined by,

$$\ell_p \equiv \|\underline{s}\|_p \equiv \left(\sum_{i=1}^{n} |s_i|^p\right)^{1/p} \qquad \text{for } p \geq 1 \tag{A.13}$$

Three of these are used:

a) the ℓ_1 norm

$$\|\underline{s}\|_1 = \sum_{i=1}^{n} |s_i| \tag{A.14}$$

b) the ℓ_2 or Euclidean norm

$$\|\underline{s}\|_2 = \left(\sum_{i=1}^{n} s_i^2\right)^{1/2} \tag{A.15}$$

c) the ℓ_∞ or maximum norm

$$\|\underline{s}\|_\infty = \max_{1 \leq i \leq n} |s_i| \tag{A.16}$$

When the ℓ_2 norm is used to norm the space R_n the result is called Euclidean space. If $n=3$ this is an analog to the "physical world."

Comment A.1: In the case of a continuous function $w(t)$, $t \in [0, T]$, The L_p norm is defined by

$$L_p \equiv \|w\|_p \equiv \left(\int_0^T |w(\tau)|^p d\tau\right)^{1/p} \qquad p \geq 1 \tag{A.17}$$

The L_∞ or supremum norm is often used.

$$L_\infty = \|w(t)\|_\infty = \sup_{t\in[0,\,T]} |w(t)| \tag{A.18}$$

Comment A.2: The norm of a matrix is defined here as the norm induced by the corresponding vector norm.

$$\|\underline{B}\| \equiv \sup_{\underline{s}\neq\underline{0}} \frac{\|\underline{B}\,\underline{s}\|}{\|\underline{s}\|} \tag{A.19}$$

It follows that

$$\|\underline{B}\|_\infty = \max_{1\leq i\leq n} \sum_{j=1}^{n} |b_{ij}|$$

$$\|\underline{B}\|_1 = \max_{1\leq j\leq n} \sum_{i=1}^{n} |b_{ij}|$$

Two useful relations now hold for any induced matrix norm.

$$\begin{aligned} \|\underline{B}\,\underline{s}\| &\leq \|\underline{B}\| \;\; \|\underline{s}\| \\ \|\underline{A}\,\underline{B}\| &\leq \|\underline{A}\| \;\; \|\underline{B}\| \end{aligned} \tag{A.20}$$

4. ANALYSIS

A sequence s_i, $i=1,2,\ldots,$ of elements of S is denoted by $\{s_i\}$. Such a sequence is called a Cauchy sequence if for any $\epsilon > 0$ there is an integer $k(\epsilon)$ such that for all k_1 and k_2 greater than $k(\epsilon)$, one has a distance function for which,

$$\rho(s_{k_1}, s_{k_2}) < \epsilon \tag{A.21}$$

A sequence has a limit s, denoted by

$$\lim_{k\to\infty} s_k = s \tag{A.22}$$

relative to a given distance function, if for any $\epsilon > 0$ there is an integer $k(\epsilon)$ such that for all $k > k(\epsilon)$

$$\rho(s_k, s) < \epsilon \tag{A.23}$$

In numerical analysis such a sequence is said to converge to a limit.

Lemma A.1: A sequence is Cauchy if and only if it has a limit. A necessary condition for convergence to a limit is that successive members of the sequence become close, i.e., the condition

$$\rho(s_{i+1}, s_i) < \epsilon \tag{A.24}$$

must be satisfied for $\epsilon > 0$.

Consider all possible sequences $\{s_i\}$ in a space S which have limits s. If all such possible limits lie within the space S, the space is said to be complete. A complete, normed space is called a Banach space. If a subset $S_1 \subset S$ has the above property of containing all its possible limit points it is a closed set.

Lemma A.2: The space R_n is complete with respect to any one of the norms ℓ_p.

Lemma A.3: In the space R_n, if a sequence converges to a limit with respect to any ℓ_p norm, it converges with respect to all ℓ_p norms (for $p \geq 1$).

The open sphere or neighborhood about a point s_0 is the set $S_r(s_0)$.

$$S_r(s_0) = \{s_0 : \|s_0 - s\| < r\} \tag{A.25}$$

The closed sphere is denoted by $\overline{S}_r(s_0)$

$$\overline{S}_r(s_0) = \{s_0 : \|s_0 - s\| \leq r\} \tag{A.26}$$

If each member s_1 of a set S_1 is the center of a neighborhood $S_r(s_1) \subset S_1$, then S_1 is called an open set.

5. STATE EQUATIONS

Given that the behavior of a given system can be modelled by a set of differential equations, there are still many ways in which the variables can be chosen and the equations written. In this thesis a set of first order differential equations, called the (vector) state equation is used.

$$\dot{\underline{x}}(t) = f(\underline{x}(t), \underline{u}(t), t) \tag{A.27}$$

The n dimensional vector function $\underline{x}(t)$ is called the state vector. The implicit dependence on time will sometimes be omitted, e.g., $\dot{\underline{x}} = f(\underline{x}, \underline{u}, t)$.

Assumption A.1: The elements of the state vector are independent. This is equivalent to requiring that the dimension n of the state space be as small as possible.

A knowledge of $u(t)$ for $t \in (t_0, t_1]$, written $u_{(t_0, t_1]}$, plus the initial value of the state $\underline{x}(t_0)$ is sufficient to determine the state $\underline{x}(t)$ for $t \in [t_0, t_1]$. In accordance with this we define the transition function $\underline{\psi}(t, u_{(t_0, t_1]}, x(t_0))$ such that

$$\underline{x}(t) = \underline{\psi}(t, u_{(t_0, t_1]}, \underline{x}(t_0)) \tag{A.28}$$

The state equation is called linear if it is of the form A.29, i.e., linear in both the state and the control

$$\dot{\underline{x}}(t) = \underline{A}(t)\,\underline{x}(t) + \underline{B}(t)\,\underline{u}(t) \tag{A.29}$$

The state equation is called linear time-invariant if it is of the form A.30.

$$\dot{\underline{x}}(t) = \underline{A}\,\underline{x}(t) + \underline{B}\,\underline{u}(t) \tag{A.30}$$

In this thesis a scalar control is used with a linear, time-invariant system.

$$\dot{\underline{x}}(t) = \underline{A}\,\underline{x}(t) + \underline{b}\,u(t) \tag{A.31}$$

Given the linear equation A.29 and the initial state $\underline{x}(t_0)$, the fundamental solution is given by,

$$\underline{x}(t) = \underline{\phi}(t, t_0)\,\underline{x}(t_0) + \underline{\phi}(t, t_0)\int_0^t \underline{\phi}^{-1}(\tau, t_0)\,\underline{B}(\tau)\,\underline{u}(\tau)\,d\tau \qquad (A.32)$$

$\underline{\phi}(t, t_0)$ is the <u>fundamental matrix</u>, the unique solution of the matrix equation

$$\dot{\underline{\phi}}(t, t_0) = \underline{A}(t)\underline{\phi}(t, t_0) \qquad (A.33)$$

subject to

$$\underline{\phi}(t_0, t_0) = \underline{I}$$

If the system is time invariant, the fundamental matrix becomes the <u>exponential matrix</u>. In this case one sets $t_0 = 0$ without loss of generality.

$$\underline{\phi}(t, 0) = \underline{e}^{\underline{A}t} = \sum_{k=0}^{\infty} \frac{1}{k!}(\underline{A}t)^k \qquad (A.34)$$

Then the equation A.30 has the fundamental solution A.35, with $\underline{x}(0) = \underline{\xi}$.

$$\underline{x}(t) = \underline{e}^{\underline{A}t}\,\underline{\xi} + \int_0^t \underline{e}^{\underline{A}(t-\tau)}\underline{b}\,u(\tau)\,d\tau \qquad (A.35)$$

6. REACHABILITY AND CONTROLLABILITY

Usually the control variable $\underline{u}(t)$ is required to lie in a given closed subset U_t of R_k at each time t. If $u(t) \epsilon U_t$, then $u(t)$ is <u>allowable at time t</u>. Define the function space of all <u>allowable control functions</u> $\underline{u}_{(t_0, t_1]}$ as $U_{(t_0, t_1]}$. That is, $\underline{u}_{(t_0, t_1]} \epsilon U_{(t_0, t_1]}$ means $\underline{u}(t)$ is an <u>allowable control</u> for all $t \epsilon (t_0, t_1]$.

A state $\underline{x}(t_1)$ is reachable at time t_1 from the state $\underline{x}(t_0)$ if there is a function $\underline{u}_{(t_0, t_1]} \in U_{(t_0, t_1]}$ such that

$$\underline{x}(t_1) = \underline{\psi}(t_1, \underline{u}_{(t_0, t_1]}, \underline{x}(t_0)) \tag{A.36}$$

The set of all reachable states at time t_1, $R(t_1, U_{(t_0, t_1]}, \underline{x}(t_0))$, is the subset of R_n which consists of all the states $\underline{x}(t_1)$ that are reachable from $\underline{x}(t_0)$ in $(t_1 - t_0)$ seconds using any allowable control function, i.e.,

$$R(t_1, U_{(t_0, t_1]}, \underline{x}(t_0)) = \{\underline{x}(t_1) : \underline{x}(t_1) = \underline{\psi}(t_1, \underline{u}_{(t_0, t_1]}, \underline{x}(t_0))$$

for some

$$\underline{u}(t_0, t_1] \in U_{(t_0, t_1]} \} \tag{A.37}$$

The set of all reachable states consists of all those reachable at any finite time $t \geq t_0$.

If there is a piecewise continuous function $\underline{u}_1(t_0, t_1]$ such that

$$\underline{\psi}(t_1, \underline{u}_1(t_0, t_1], \underline{x}(t_0)) = \underline{0} \tag{A.38}$$

$$\text{for some } t_1 \geq t_0$$

then the state $\underline{x}(t_0)$ is controllable at t_0.

If every state $\underline{x}(t_0)$ is controllable for every time t_0, the system is completely controllable (or just controllable).

Given the linear, time invariant system A.30 the condition for controllability takes a simple form: Define the controllability matrix $\underline{G}$, whose k^{th} column is the vector $\underline{A}^{k-1}\underline{b}$.

$$\underline{G} = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ \underline{b} & \underline{Ab} & \underline{A^2 b} & \cdots \underline{A^{n-1} b} \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \tag{A.39}$$

The condition for controllability is then

$$\det \underline{G} \neq 0 \tag{A.40}$$

7. THE MINIMUM PRINCIPLE

The theorem is stated only for the fixed time, fixed end point problem considered in this thesis. More general formulations are given in Pontryagin, et al.[84] or Athans and Falb.[4]

Given the fixed time, fixed end point plant of Chapter II,

$$\dot{\underline{x}}(t) = \underline{f}(\underline{x}(t), \underline{u}(t), t) \tag{A.41}$$

$$\underline{x}(t_0) = \underline{\xi}$$

$$\underline{x}(t_1) = \underline{\theta}$$

$$t_1 \text{ fixed}$$

and a cost functional of integral type.

$$J(\underline{x}(t), \underline{u}(t), t) = \int_{t_0}^{t_1} L(\underline{x}(\tau), \underline{u}(\tau), \tau) d\tau \tag{A.42}$$

assume the set of allowable controls is not time varying.

$$u(t) \in U \quad \text{for all} \quad t \in [t_0, t_1]$$

Assume that each element of

a) the functions

$$L(\underline{x}, \underline{u}, t) \quad \text{and} \quad \frac{\partial L}{\partial t} (\underline{x}, \underline{u}, t)$$

b) the vectors

$$\underline{f}(\underline{x}, \underline{u}, t), \ \frac{\partial \underline{f}}{\partial t} (\underline{x}, \underline{u}, t), \ \frac{\partial L}{\partial \underline{x}} (\underline{x}, \underline{u}, t)$$

c) and the matrix

$$\frac{\partial \underline{f}}{\partial \underline{x}} (\underline{x}, \underline{u}, t)$$

is continuous on the subset $R_n \times U \times (t_0, t_1)$ of its domain space.

Define the costate vector $\underline{p}(t)$ and the Hamiltonian $H(\underline{x}(t), \underline{u}(t), \underline{p}(t), t)$ such that

$$H(\underline{x}(t), \underline{u}(t), \underline{p}(t), t) = L(\underline{x}(t), \underline{u}(t), t) + \underline{p}'(t)\, f(\underline{x}(t), \underline{u}(t), t) \tag{A.43}$$

and the canonical (or Hamiltonian) equations are,

$$\begin{aligned} \dot{\underline{x}}(t) &= \frac{\partial H}{\partial \underline{p}}(\underline{x}, \underline{u}, \underline{p}, t) = f(\underline{x}, \underline{u}, t) \\ \dot{\underline{p}}(t) &= -\frac{\partial H}{\partial \underline{x}}(\underline{x}, \underline{u}, \underline{p}, t) = -\frac{\partial L}{\partial \underline{x}}(\underline{x}, \underline{u}, t) - \left(\frac{\partial f}{\partial \underline{x}}(\underline{x}, \underline{u}, t)\right)' \underline{p} \end{aligned} \tag{A.44}$$

Let $\underline{u}^*(t)$ be an allowable control such that the corresponding trajectory $\underline{x}^*(t)$ begins at the point $\underline{\xi}$ at time t_0 and is at the point $\underline{\theta}$ at time t_1.

Theorem A.1: In order for $\underline{u}^*(t)$ to be optimal, it is necessary that there exist a costate function $\underline{p}^*(t)$ such that:

a) $\underline{p}^*(t)$ corresponds to $\underline{u}^*(t)$ and $\underline{x}^*(t)$ as a solution of the canonical equations.

$$\begin{aligned} \dot{\underline{x}}^*(t) &= \frac{\partial H}{\partial \underline{p}}(\underline{x}^*(t), \underline{p}^*(t), \underline{u}^*(t), t) \\ \dot{\underline{p}}^*(t) &= -\frac{\partial H}{\partial \underline{x}}(\underline{x}^*(t), \underline{p}^*(t), \underline{u}^*(t), t) \end{aligned} \tag{A.45}$$

b) for all $t \in [t_0, t_1]$ the Hamiltonian has an absolute minimum as a function of $u(t)$ over U. That is

$$H(\underline{x}^*(t), \underline{u}^*(t), \underline{p}^*(t), t) \leq H(\underline{x}^*(t), u(t), p^*(t), t) \quad \text{for all } u(t) \in U \tag{A.46}$$

APPENDIX B

NEWTON'S METHOD IN FUNCTION SPACE

The purpose of this appendix is to define some basic notions in function space, and to present some results on Newton's method. For more general and complete treatments see e.g., Kantorovich and Akilov,[49] Kolmogorov and Fomin,[59] the paper by Moore in Anselone,[2] or the chapter by Antosiewicz and Rheinbolt in Todd.[3]

1. FUNCTION SPACE

Consider an ordinary function $g(t)$. This is a mapping from R_1 into R_1, written $g(t) : R_1 \rightarrow R_1$ A set of such functions can be considered as a space G of functions, or a <u>function space</u>. Then each element of G is a function $g^j(t)$.* The space C defined in Appendix A.2 is an example of a function space. One could equally well have a space each element of which is a vector function, i.e., $\underline{g}(t) : R_1 \rightarrow R_m$. Or the argument of each element could be a vector also, i.e., $\underline{g}(\underline{y}) : R_\ell \rightarrow R_m$. Finally, using the function space G one could define a functional on the function space.

Example: let $$J(u(t)) = \int_{t_0}^{t_1} |u(\tau)| d\tau$$

Then J is a functional, such that,

$$J : U \rightarrow R_1$$

---

* A superscript index will be used to denote a particular element of the set, in order to agree with the notation used in this thesis for Newton's method.

The definitions of metric space, linear space, normed space, and convergence given in Appendix A apply equally well to function spaces. An example of a norm in the space C of continuous functions is the L_p norm defined in Section A.3.

2. DERIVATIVES

The derivative of an operator in function space is defined in a way rather analogous to the derivative of an ordinary function in R_1. The first derivative is needed to carry out Newton's method.

Let Y and Z be two Banach spaces. Let $P(\cdot)$ be an operator mapping an open subset Y_1 of Y into a subset Z_1 of Z. Let y^0 be a fixed element of the subset Y_1. Then $y^0 \in Y_1 \subset Y$. Suppose there exists a linear operator $P^{(1)}[y^0](\cdot)$ such that for every $y \in Y$

$$P^{(1)}[y^0](y) \triangleq \lim_{\epsilon \to 0} \frac{1}{\epsilon} \{P(y^0 + \epsilon y) - P(y^0)\} \qquad (B.1)$$

Then the linear operator $P^{(1)}[y^0](\cdot)$ is called the <u>derivative</u> of the operator P evaluated at the element y^0. This derivative is often called the weak or Gateaux derivative, and the element $P^{(1)}[y^0](y)$ is called the Gateaux differential. If in addition the convergence is uniform for all $y \in Y$ with $\|y\| = 1$, then the operator P is <u>differentiable</u> at the element y^0. In this case the operator $P^{(1)}$ is called the strong or Frechet derivative (or sometimes just the derivative for short).

In Theorems B.1 and B.2 on the convergence of Newton's method the second derivative of an operator $P(\cdot)$ is used. This is defined quite naturally as the derivative of the first derivative operator (when this operation exists) and is a bilinear operator. Suppose there exists a bilinear operator $P^{(2)}[y^0](\cdot,\cdot)$ such that for every $v, w, \in Y$

$$P^{(2)}[y^0](v,w) = \lim_{\epsilon \to 0} \frac{1}{\epsilon} \{P^{(1)}[y^0 + \epsilon w]v - P^{(1)}[y^0]v\} \qquad (B.2)$$

The bilinear operator $P^{(2)}[y^0](\cdot,\cdot)$ is called the weak second derivative of the operator $P(\cdot)$ evaluated at the element y^0. If, in addition, the convergence is uniform for all v, w, ϵ, Y with $\|v\| = \|w\| = 1$, then the operator $P(\cdot)$ is twice differentiable at the element y^0. $P^{(2)}[y^0](\cdot,\cdot)$ is then called the strong second derivative (or sometimes simply the second derivative).

Note that the operator $P^{(1)}[y^0 + \epsilon w](\cdot)$ above is the first derivative of the operator P, evaluated at the element $y^0 + \epsilon w$. The operator $P^{(1)}[y^0](\cdot)$ is the first derivative of the operator P evaluated at the element y^0.

The norm of the second derivative operator $P^{(2)}[y^0]$ is defined as,

$$\| P^{(2)}[y^0] \| = \sup_{\substack{\| v \| = 1 \\ \| w \| = 1}} \| P^{(2)}[y^0](v, w) \|$$

3. NEWTON'S METHOD

Assume the same operator $P(\cdot)$ as in Section B.2, and suppose that the zero element $\odot$ is part of the subset Z_1. That is, that there exists an element $y^* \in Y_1$ (a <u>zero</u> of the operator P) such that

$$P(y^*) = \odot \tag{B.3}$$

Suppose, further, that $P(\cdot)$ has a continuous first derivative in Y_1. Let y^0 be an element near y^* such that $y^0 \in Y_1$. A Taylor series expansion about y^0 yields

$$\odot = P(y^*) = P(y^0) + P^{(1)}[y^0](y^* - y^0) + r \tag{B.4}$$

where r represents the higher order terms.

Assume the first derivative operator $P^{(1)}[y^0](\cdot)$ has an inverse denoted by $[P^{(1)}[y^0]]^{-1}(\cdot)$. A first order estimate of y^* can be obtained from Eq. B.4 by dropping the higher order terms.

$$y^* \approx y^0 - [P^{(1)}[y^0]]^{-1} P(y^0) \tag{B.5}$$

For convenience define

$$\Gamma_0 = [P^{(1)}[y^0]]^{-1} \tag{B.6}$$

<u>Newton's method</u> consists of applying the estimate (B.5) recursively to generate a sequence of elements $\{y^i\}$.

$$y^1 = y^0 - \Gamma_0 \, P(y^0)$$

$$y^2 = y^1 - \Gamma_1 \, P(y^1)$$

and in general, at the ith step

$$y^{i+1} = y^i - \Gamma_i P(y^i) \tag{B.7}$$

The sequence of elements $\{y^i\}$ is said to converge if

$$\lim_{i \to \infty} y^i = y^* \tag{B.8}$$

Comment B.1: If the operator $P(\cdot)$ is a real function of a real argument (an ordinary function), then Eq. B.7 becomes

$$y^{i+1} = y^i - \frac{P(y^i)}{P^{(1)}(y^i)}$$

which leads to Newton's method in its ordinary form (also called the Method of Tangents).

Comment B.2: For completeness, the modified Newton's method is included here. If a simpler recursive scheme is desired, one can compute the inverse first derivative operator Γ only once and keep using the original one Γ_0. This leads to the recursive relation

$$y^{i+1} = y^i - \Gamma_0 P(y^i) \tag{B.9}$$

Equation B.9 involves less computation than Eq. B.7, but the rate of convergence is generally slower, being usually exponential in nature.

4. SUFFICIENT CONDITIONS FOR CONVERGENCE

There have been a number of theorems giving sufficient conditions for the convergence of Newton's method, including recent ones by Kalaba,[47] by McGill and Kenneth,[67] and by Kantorovich.[48] The theorem by Kantorovich was found to be very general and powerful, and is the one stated below. Several forms of this theorem can be found in Kantorovich and Akilov,[49] Chapter XVIII, together with the proof which is somewhat lengthy and will not be reproduced here.

For convenience, the definitions of open and closed spheres or neighborhoods are repeated here from Appendix A.4.

$$S_{r_0}(y^0) = \{y: \| y-y^0 \| < r_0\}$$

and

$$\overline{S}_{r_1}(y^0) = \{y: \| y-y^0 \| \leq r_1\}$$

Let the operator $P(y)$ be defined as before on an open set $Y_1 \subset Y$ and let the operator equation

$$P(y) = \odot \tag{B.10}$$

be given. The object is to find a solution y^* satisfying Eq. B.10, i.e., a zero of the operator $P(\cdot)$. A fixed element y^0 is given as a starting guess (initial approximation).

Theorem B.1: Suppose that:

1. The second derivative operator $P^{(2)}[y](\cdot,\cdot)$ exists and is continuous on the set Y_1.

2. The first derivative inverse operator Γ_0 exists

3. $\| \Gamma_0 P(y^0) \| \leq \beta_0$ (B.11)

4. $\| \Gamma_0 P^{(2)}[y] \| \leq \beta$ for all $y \in Y_1$ (B.12)

5. $h = \beta_0 \beta \leq 1/2$ (B.13)

6. $Y_1 \supset \overline{S}_{r_0}(y^0)$ (B.14)

where $$r_0 = \frac{1 - \sqrt{1 - 2h}}{\beta} \tag{B.15}$$

Then:

1. There is a solution $y^* \in \overline{S}_{r_0}(y^0)$ such that

$$P(y^*) = \odot$$

2. This solution is unique in the set $Y_1 \cap S_{r_1}(y^0)$, where

$$r_1 = \frac{1 + \sqrt{1 - 2h}}{\beta} \tag{B.16}$$

3. Newton's method converges to the solution y^*.

4. The rate of convergence is characterized by the inequality

$$\| y^* - y^i \| \leq \frac{(2h)^{2^i}}{\beta 2^i} \qquad \text{(B.17)}$$

It is actually sufficient to evaluate the second derivative operator only over the neighborhood $S_{r_0}(y^0)$. The difficulty is that the radius r_0 is not known in advance. However, one approach is to choose a radius r and check to see if the conditions are fulfilled. This leads to Theorem B.2, which is the form of the theorem given by Kantorovich and Akilov.[49] Those parts which are changed from Theorem B.1 are marked by bracketed numbers.

Theorem B.2: Suppose that:

1. The second derivative operator $P^{(2)}[y](\cdot,\cdot)$ exists and is continuous in the neighborhood $S_r(y^0)$. This obviously requires

$$Y_1 \supset S_r(y^0)$$

2. The first derivative inverse operator Γ_0 exists.

3. $\| \Gamma_0 P(y^0) \| \leq \beta_0$

4. $\| \Gamma_0 P^{(2)}[y] \| \leq \beta$ for all $y \in S_r(y^0)$ (B.18)

5. $h = \beta_0 \beta \leq 1/2$

6. $r \geq r_0 = \dfrac{1 - \sqrt{1 - 2h}}{\beta}$

Then:

1. There is a solution $y^* \in \overline{S}_{r_0}(y^0)$ such that

$$P(y^*) = \odot$$

2. Let

$$r_1 = \min\left(r, \frac{1 + \sqrt{1 - 2h}}{\beta}\right) \qquad \text{(B.19)}$$

Then the solution y^* is unique in the sphere $S_{r_1}(y^0)$.

3. Newton's method converges to the solution y^*.

4. The rate of convergence is characterized by the inequality

$$\| y^* - y^i \| \leq \frac{(2h)^{2^i}}{\beta 2^i}$$

Comment B.3: The inequality (B.17) leads to the property of "Asymptotically Quadratic" convergence. This means that the quantity $y^* - y^i$ is approximately squared at each step. Another way of expressing this is by noting that the number of correct significant digits in y^i approximately doubles at each step. "Asymptotic" means that the property may not begin to appear until after some iterations have already taken place.

Comment B.4: If the operator P(y) is not defined in the entire neighborhood $S_{r_0}(y^0)$ two things can happen. First, although nearby conditions point toward a solution y^*, there may be none within the set Y_1 on which P(y) is defined. Second, even if the solution does exist, Newton's method may at some step y^i go outside the domain of definition Y_1 and hence fail to reach the solution.

5. EXAMPLES

Normally one would expect from Theorem B.1 that in a given problem there would be a set of initial guesses y^0 for which Newton's method would converge (the Region of Convergence of Chapter II.), and other initial guesses for which it would not converge. The computer examples studied appear to have this type of behavior.

In order to show some other types of behaviour which can result, three simple scalar examples are shown below.

Example 1

$$P(y) = y^3$$ See Figure B.1

The convergence theorem yields

$$h = \sup_y \frac{P(y^0)P^{(2)}[y]}{[P^{(1)}[y^0]]^2} = \sup_y \frac{(y^0)^3 6y}{9(y^0)^4} = \sup_y \left[2/3 \frac{y}{y^0} \right]$$

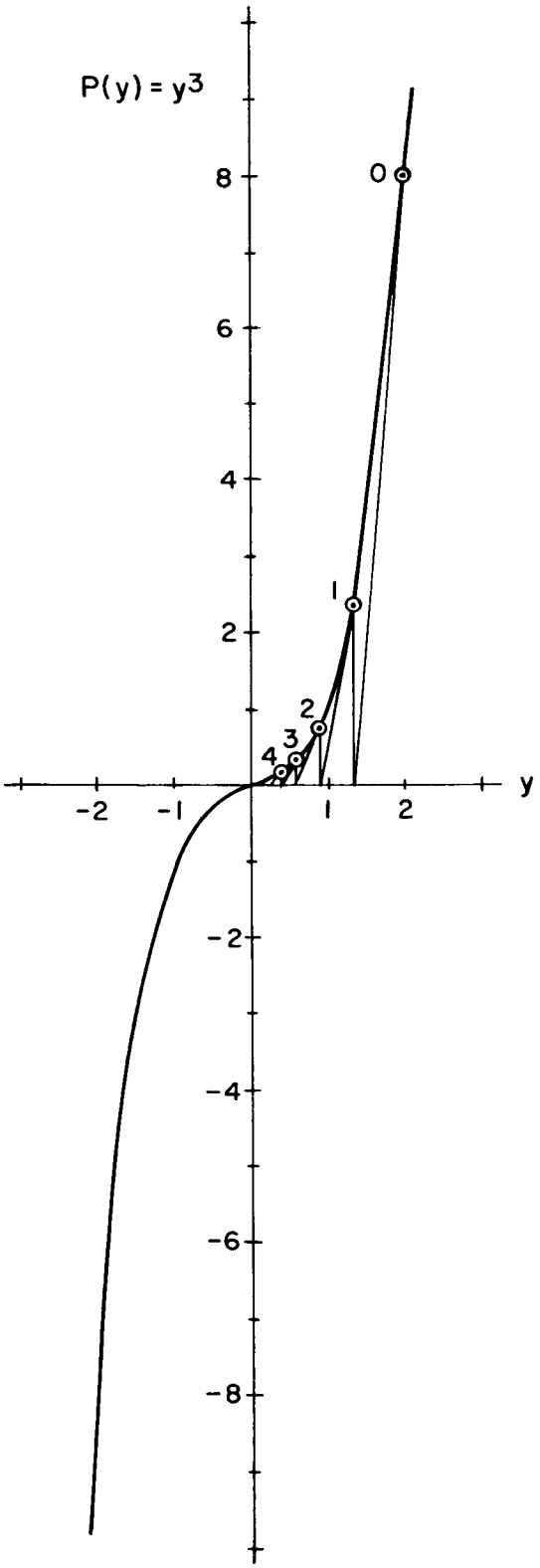


Fig. B.1 Example 1 for Newton's Method

Clearly the convergence theorem can never be satisfied for any y^0, since even with Theorem B.2 the sup operation is over values of y on both sides of y^0. However, Newton's method

$$y^{i+1} = y^i - 1/3\ y^i$$

converges for every initial guess y^0, though at a very slow rate.

Example 2

$$P(y) = |y|^{1/2}\ \mathrm{sgn}(y) \qquad \text{See Figure B.2}$$

The convergence theorem yields

$$h = \sup_y \left(\frac{|y|}{|y^0|} \right)^{3/2}$$

which is rather similar to that found in Example 1. However, Newton's method yields

$$\begin{aligned} y^{i+1} &= y^i - 2\ |y^i|\ \mathrm{sgn}\ y^i \\ &= -\ y^i \end{aligned}$$

So this example does not converge for any initial guess y^0 (except $y^0 = 0$). The second derivative becomes large much faster than the first as the root is approached, so h is always $> 1/2$.

Example 3

$$P(y) = y^2 + \epsilon \qquad \text{See Figure B.3}$$

The convergence theorem yields

$$h = 1/2 + \frac{\epsilon}{2(y^0)^2}$$

And Newton's method is

$$y^{i+1} = \frac{1}{2}\ y^i - \frac{\epsilon}{2y^i}$$

This function is very special because the sufficient condition of Kantorovich is also necessary. For $\epsilon=0$, the theorem yields $h = 1/2$ and Newton's method just converges. The rate of convergence in this case is only exponential (error is reduced by 1/2 at each step). For any $\epsilon > 0$ there is no longer a root, $h > 1/2$, and Newton's method finally oscillates instead of converging. For any $\epsilon < 0$ the theorem

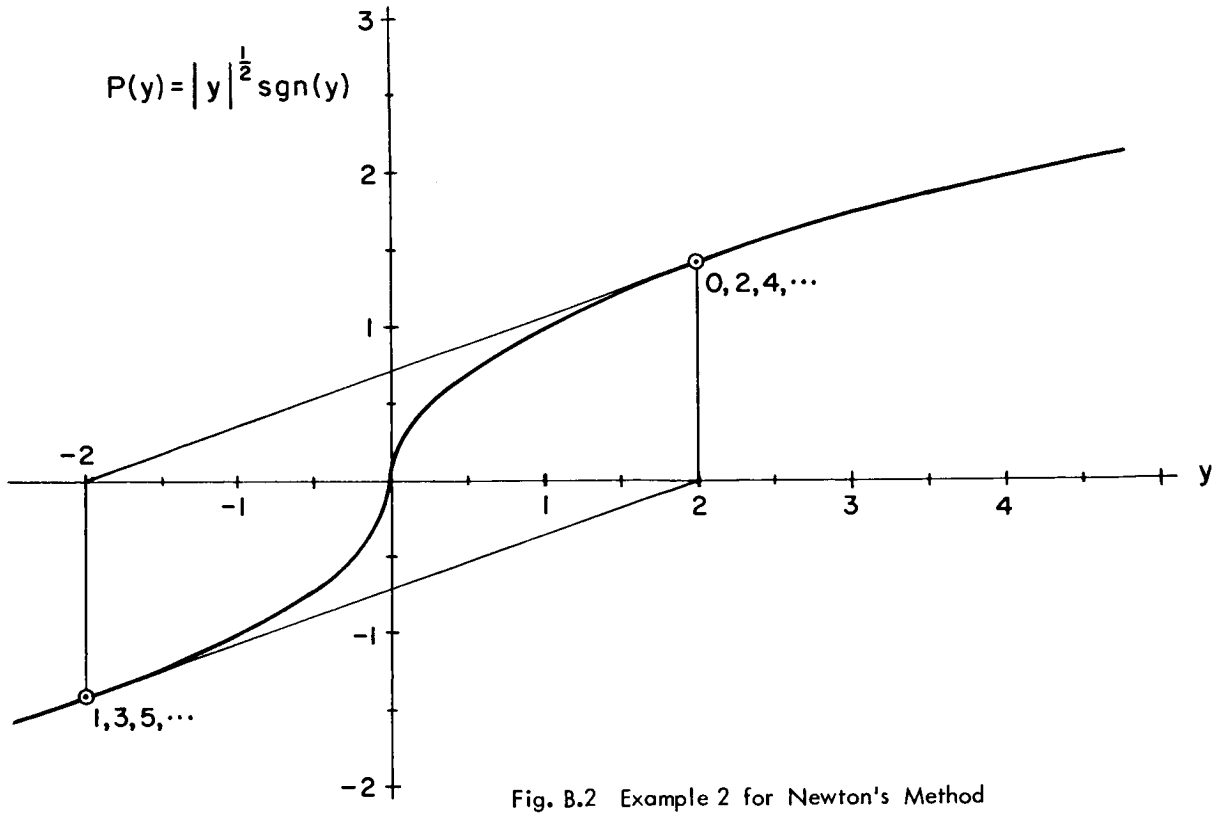


Fig. B.2 Example 2 for Newton's Method

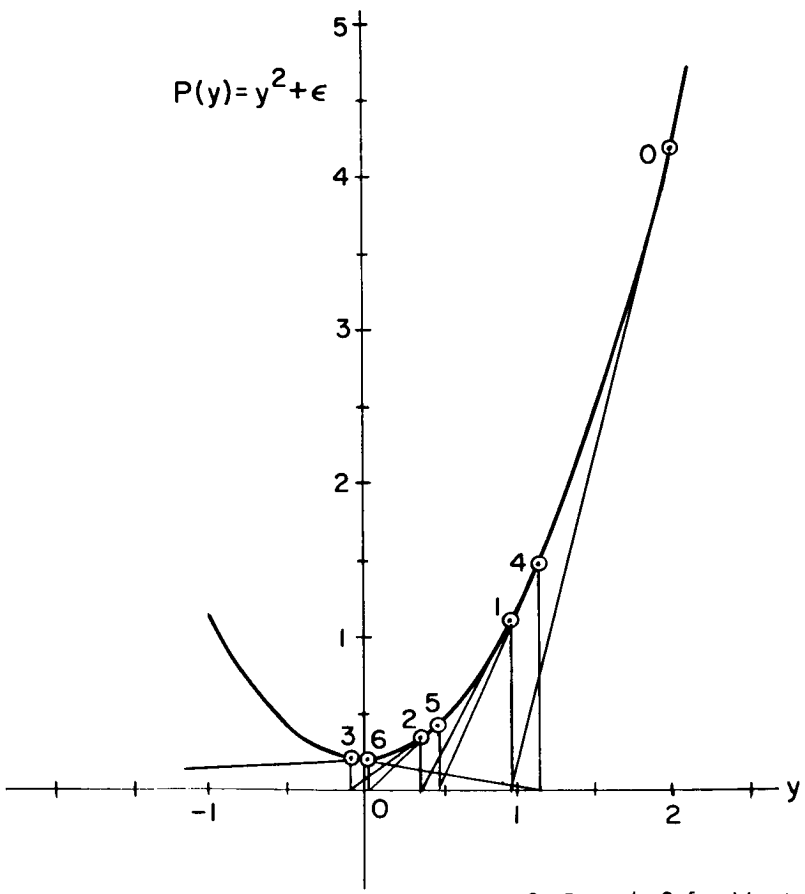


Fig. B.3 Example 3 for Newton's Method

yields $h < 1/2$ and Newton's method converges with the asymptotic quadratic convergence rate showing up as $\| y - y^* \|$ becomes small.

As these examples show, the properties of the theorem do not necessarily follow if the conditions are not fulfilled. Thus Newton's method is not guaranteed to converge at a quadratic rate when it converges. A few of the very difficult and impossible computer examples illustrated this. Also Newton's method may not converge no matter how close to the root one starts. As shown in Chapter III, Sec. D, for the class of optimal control problems studied this second type of peculiar behaviour cannot occur. (Primarily since the second derivative operator is bounded.)

APPENDIX C

DETAILS OF COMPUTER RUNS

Run No.	a_0	η_1	a_1	No. of Iter.	η_2	a_2	No. of Iter.	η_3	No. of Iter.	η_4	No. of Iter.	η_5	No. of Iter.	η_6	No. of Iter.	η_7	No. of Iter.
1	.0888	.0667	.02	2	.177	0	4	.47	4	1.25	3	3.31	2	8.8	2		
2																	
3	.176	.0734	.0963	2	.195	0	7	.356	9	.65	7	1.73	6	4.5	3		
4	.14	.20	0	3	.366	0	6	.668	6	1.22	5						
5	.1273	.20	0	3	.366	0	5	.97	3								
6	.0706	.1012	0	3	.269	0	6	.714	4	1.89	3						
7	.0835	.129	0	4	.235	0	5	.624	5	1.65	4	4.4	3				
8	.044	.097	0	4	.26	0	4	.68	4	1.81	2	4.82	3				
9	.107	.1202	.0034	3	.238	0	5	.473	6	.94	5	1.86	4			7.3	2
10	.006	.009	0	3	.018	0	4	.035	4	.07	4	.14	3	.28	3	.55	2
11	.174	.315	0	3	.626	0	6	1.24	5	2.46	4	4.9	3	9.7	4		
12	.174	.315	0	3	.626	0	6	1.24	4	2.46	3	4.9	3	9.7	2		
13	.279	.308	.021	4	.610	0	4	1.21	6	2.40	4	4.77	3	9.46	4		
14																	
15	.159	.036	0	4	.608	0	5	1.20	4	2.4	3	4.7	3	9.4	2		
16	.198	.284	0	5	.564	0	7	1.12	6	2.22	4	4.4	3	8.7	3		
17	.211	.305	0	6	.605	0	7	1.2	4	2.4	3	4.7	2	9.4	2		
18																	
19	.213	.303	0	8	.602	0	9	1.2	5	2.37	4	4.7	4	9.3	3		
20																	
21	.217	.207	.016	4	.411	0	7	.816	5	1.62	4	3.21	3	6.37	4		
22	1.31	.29	.96	2	.58	.26	4	1.15	9	2.29	6	4.54	6	9.01	5		
23	.11	.061	.054	2	.163	0	5	.30	8	.54	6	1.44	6	3.8	4		
24	.030	.106	0	4	.28	0	4	.75	3	2.0	4	3.63	4	9.6	4		
25	.054	.036	.021	2	.097	0	3	.177	6	.324	6	.86	6	2.3	4	6.1	5
26	.82	.24	.54	2	.49	0	8	.96	8	1.92	5	3.8	4	7.5	3		
27	1.34	.22	1.09	2	.436	.56	3	.86	5	1.72	6	3.4	5	6.7	4		
28	.246	.38	0	10	.76	0	6	1.50	4	2.97	3	5.90	6				
29	.994	.256	.697	2	.508	.081	4	1.0	8	2.0	3	3.97	4	7.9	9		
30	.125	.205	.062	4	.406	0	3	.806	5	1.60	4	3.17	4	6.29	3		
31	.187	.133	.0566	3	.265	0	7	.525	10	1.04	7	2.07	8	4.10	11	8.14	10
32	.104	.252	0	5	.436	0	4	.752	4	1.30	3	2.24	3	3.87	4	6.68	4
33	.169	.280	0	4	.484	0	6	.835	5	1.44	4	2.49	4	4.29	3	7.41	4

```
C       MAIN PROGRAM
        DIMENSION A(10,10),B(10),Q(10,101),PO(10),EMAT(10,10),AT(10,10),TH
       1ETA(10),ZETA(10),Z(10),BIG(10),AIDENT(10,10),PERMU(10),WI(10,10),C
       12(10,10),W(10,10),QSUP(10),EFETA(10),EAT(10,10)
        COMMON Q,PO,A,B,AT,Z,N,M,DELTA,DELSR,ETA,AIDENT,PERMU,L,WI,C2
       1,W,ALPHA,ALPH1,ICON,EFETA,?NORM,ETNORM,ZETA,EAT
4       READ 100,N,T
        CALL RSCLCK
         PRINT 109
        PRINT 110,N,T
        READ 101,(ZETA(I),I=1,N)
        PRINT 111,(ZETA(I),I=1,N)
        DO 2 I=1,N
        READ 101,(A(I,J),J=1,N)
 2      PRINT 111,(A(I,J),J=1,N)
        READ 101,(B(I),I=1,N)
        PRINT 111,(B(I),I=1,N)
        READ 102,EPS,AMAX,EPMTX,ALPT,M,ICHO,KPETA
        PRINT 102,EPS,AMAX,EPMTX,ALPT,M,ICHO,KPETA
        READ 101,(THETA(I),I=1,N)
        PRINT 111,(THETA(I),I=1,N)
        DELTA=T/FLOATF(M)
        CALL QMAT(QSUP,EPMTX)
        PRINT 114
        DO 10 I=1,N
 10     PRINT 111,(AT(I,J),J=1,N)
        CALL INIT (THETA,ICHO)
        L=L
        GO TO(6,7,8),L
7       PRINT 118
        DO 15 I=1,N
15      PRINT 111,(W(I,J),J=1,N)
        GO TO 4
8       PRINT 119
        GO TO 4
6       PRINT 117
        DO 18 I=1,N
18      PRINT 111,(W(I,J),J=1,N)
        PRINT 122,ALPHA
        PRINT 115
        PRINT 111,(PO(I),I=1,N)
        CALL STOPCL(I)
        PRINT 103,I
        CALL START(QSUP)
        ICOUNT =-1
        PRINT 116,ETA,ALPH1
        IF(ICHO-3)3,21,21
3       CALL ITER (EPS)
        CALL STOPCL(I)
        PRINT 103,I
        L=L
        GO TO(11,12,13),L
13      PRINT 121
        GO TO 4
12      PRINT 120
        DO 16 I=1,N
16      PRINT 111,(C2(I,J),J=1,N)
11      PRINT 115
        PRINT 111,(PO(I),I=1,N)
        IF(XMODF(ICHO,3)-1)20,20,19
 19     IF(ICON)20,20,17
 17     CALL SSTRAJ
20      CALL CHGETA(ICOUNT,ALPT,KPETA)
        PRINT 116,ETA,ALPH1
        IF(ETA-AMAX)22,22,24
22      IF(ICHO-3)3,21,21
21      CALL CKCON(H)
        PRINT 123,H
        IF(ICHO-6)23,3,3
```

```
23    ICON=-1
      IF(H-.5)3,3,20
 24   CALL STOPCL(I)
      PRINT 103,I
      IF(XMODF(ICHO,3)-1)4,25,25
 25   PRINT 124
      CALL ETABIG (EPS)
      PRINT 115
      PRINT 111,(PO(I),I=1,N)
      GO TO 4
100   FORMAT(I5,F15.8)
 101  FORMAT (5E15.8)
 102  FORMAT (1H ,E9.3,3E10.4,3I5.)
 103  FORMAT (80X7HTIME IS,I6,19H 60THS OF A SECOND. )
109   FORMAT (1H1,20X 52HOPTIMAL CONTROL APPROXIMATION PROGRAM - MINIMUM
     1 FUEL///)
110   FORMAT (I5,43H  DIMENSION STATE SPACE    TERMINAL TIME =  ,F12.4,9
     1H  SECONDS///)
 111  FORMAT (8E15.8)
114   FORMAT (///49H  THE NEGATIVE TIME EXPONENTIAL MATRIX, E-AT, IS  )
115   FORMAT (///23H  THE COSTATE GUESS IS )
116   FORMAT (//22H  THE VALUE OF ETA IS ,E14.8,23H, THE VALUE OF ALPH1
     1IS ,E14.8)
117   FORMAT (///35H      THE CONTROLLABILITY MATRIX IS, )
118   FORMAT (114H THE INVERSE CONTROLLABILITY MATRIX IS INACCURATE.  OV
     1ERFLOW HAS OCCURRED.  THE INVERSE CONTROLLABILITY MATRIX IS,  )
119   FORMAT (30X86HTHE CONTROLLABILITY MATRIX HAS NO INVERSE.  $$$$ A
     1THING MUST BE WRONG WITH THE DATA. )
120   FORMAT (20X91H THE FIRST DERIVATIVE IS TOO FLAT AT THIS POINT, BUT
     1WE-LL USE IT ANYWAY.  THE C2I MATRIX IS  )
121   FORMAT (30X81H THE INVERSE OF THE FIRST DERIVATIVE FUNCTION FAILS
     1TO EXIST AT THIS POINT.  QUIT  )
122   FORMAT (///10X24HA LINEAR PIECE, OF SLOPE, E14.8, 34HHAS BEEN REMO
     1VED FROM THE CONTROL.  )
123   FORMAT (//40X61H  THE CONVERGENCE THEOREM OF KANTOROVICH YIELDS A
     1VALUE OF H= ,E14.8)
124   FORMAT (//32H NOW TRY FOR THE EXACT SOLUTION. )
      END



      SUBROUTINE QMAT(QSUP,EPMTX)
C     COMPUTATION OF THE MATRIX EXPONENTIAL AND THE Q VECTORS
      DIMENSION A(10,10),B(10),Q(10,101),PO(10),EMAT(10,10),AT(10,10),E(
     110),D(10),        AIDENT(10,10),PERMU(10),Z(10),WI(10,10),C2(10,10)
     1),W(10,10),QSUP(10),EFETA(10),ZETA(10),EAT(10,10)
      COMMON Q,PO,A,B,AT,Z,N,M,DELTA,DELSR,ETA,AIDENT,PERMU,L,WI,C2
     1,W,ALPHA,ALPH1,ICON,EFETA,ZNORM,ETNORM,ZETA,EAT
C     COMPUTE AT MATRIX AND ENTER IDENTITY MATRIX
      DET=1.
      DO 151 I=1,N
      DO 200 J=1,N
      AT(I,J)=-A(I,J)*DELTA
200   EMAT(I,J)=0.
151   EMAT(I,I)=1.0
C     COMPUTE MATRIX EXPONENTIAL
      DO 300 I=1,N
      DO 224 J=1,N
 224  E(J)=EMAT(I,J)
      G=1.0
225   DO 270 J=1,N
      C=0.0
      DO 250 K=1,N
 250  C=(E(K)*(AT(K,J)/G))+C
      D(J)=C
 270  CONTINUE
      G=G+1.0
      DO 276 J=1,N
      EMAT(I,J)=EMAT(I,J)+D(J)
      E(J)=D(J)
```

```
 276  CONTINUE
C     TEST ON CONVERGENCE OF MATRIX EXPONENTIAL
      IF(G-5.)225,272,272
 272  DO 280 J=1,N
      IF(EMAT(I,J))273,275,273
 273  RATIO=ABSF(D(J)/EMAT(I,J))
      IF(RATIO-EPMTX)275,275,225
 275  AIDENT(I,J)=0.
 280  EAT(I,J)=EMAT(I,J)
 300  AIDENT(I,I)=1.
      L=XSIMEQF(10,N,N,EAT,AIDENT,DET,PERMU)
      IF(L-2)199,198,198
 198  PRINT 108
 108  FORMAT (38X62HHOW CAN THE FUNDAMENTAL MATRIX FAIL TO HAVE AN INVER
     1SE.  TILT.  )
C     COMPUTE THE Q VECTOR FOR EACH TIME-ENTER IDENTITY MATRIX TO START
 199  DO 306 I=1,N
      DO 301 J=1,N
301   AT(I,J)=0.
      QSUP(I)=ABSF(B(I))
      Q(I,1)=B(I)
306   AT(I,I)=1.
      MI=M+1
      DO 313 K=2,MI
C     ADVANCE TO THE NEXT TIME INCREMENT.
      DO 312 I=1,N
      DO 310 J=1,N
      C=0.
      DO 311 L=1,N
 311  C=C+AT(I,L)*EMAT(L,J)
310   D(J)=C
      DO 312 J=1,N
312   AT(I,J)=D(J)
C     COMPUTE THE Q VECTOR
      DO 313 I=1,N
      C=0.
      DO 309 J=1,N
 309  C=C+AT(I,J)*B(J)
      Q(I,K)=C
313   QSUP(I)=MAX1F(ABSF(Q(I,K)),QSUP(I))
C     THE AT(I,J)MATRIX IS NOW EXP(-AT).
      RETURN
      END



      SUBROUTINE INIT (THETA,ICHO)
C     COMPUTE THE CONTROLLABILITY MATRIX AND THE INITIAL COSTATE GUESS.
      DIMENSION THETA(10),W(10,10),Q(10,101),PO(10),A(10,10),B(10),AT(10
     1,10),WI(10,10),Z(10),AIDENT(10,10),PERMU(10),C2(10,10),ZETA(10)
     2,QSUP(10),EFETA(10),EAT(10,10)
      COMMON Q,PO,A,B,AT,Z,N,M,DELTA,DELSR,ETA,AIDENT,PERMU,L,WI,C2
     1,W,ALPHA,ALPH1,ICON,EFETA,ENORM,ETNORM,ZETA,EAT
C     COMPUTE THE CONTROLLABILITY MATRIX
      DET=1.
      DO 10 I=1,N
      DO 10 J=1,N
 10   W(I,J)=.5*(Q(I,1)*Q(J,1)+Q(I,M+1)*Q(J,M+1))
      DO 20 I=1,N
      DO 19 J=1,N
      DO 18 K=2,M
18    W(I,J)=W(I,J)+Q(I,K)*Q(J,K)
      W(I,J)=DELTA*W(I,J)
      WI(I,J)=W(I,J)
19    AIDENT(I,J)=0.
20    AIDENT(I,I)=1.0
      L=XSIMEQF(10,N,N,WI,AIDENT,DET,PERMU)
C     COMPUTE THE EQUIVALENT INITIAL CONDITION
```

```
      IF(ICHO-10)13,1,1
 1    ICHO=ICHO-10
      MI=M+1
      ARGA=0.
      ARGB=0.
      DO 2 I=1,N
      PO(I)=ZETA(I)
      ARGA=ARGA+Q(I,1)*PO(I)
 2    ARGB=ARGB+Q(I,MI)*PO(I)
      PEN=0.
      FB=0.
      F=0.
      IF(ABSF(ARGA)-1.)4,3,3
 3    F=SIGNF(1.,ARGA)*DELTA
      PEN=PEN+DELTA/2.
 4    IF(ABSF(ARGB)-1.)6,5,5
 5    FB=SIGNF(1.,ARGB)*DELTA
      PEN=PEN+DELTA/2.
 6    DO 7 I=1,N
 7    ZETA(I)=.5*(Q(I,1)*F+Q(I,MI)*FB)
      DO 12 K=2,M
      ARG=0.
      DO 8 I=1,N
 8    ARG=ARG+Q(I,K)*PO(I)
      F=0.
      IF(ABSF(ARG)-1.)11,9,9
 9    F=SIGNF(1.,ARG)*DELTA
      PEN=PEN+DELTA
 11   DO 12 I=1,N
 12   ZETA(I)=ZETA(I)+Q(I,K)*F
      PRINT 102
      PRINT 101,(PO(I),I=1,N)
      PRINT 100
      PRINT 101,(ZETA(I),I=1,N)
      PRINT 103,PEN
C     TAKE CARE OF NONZERO THETA.
 13   DO 23 I=1,N
      C=0.
      DO 22 J=1,N
 22   C=C+AT(I,J)*THETA(J)
 23   Z(I)=ZETA(I)-C

      PONORM=0.
      DO 30 I=1,N
      D=0.
      DO 24 J=1,N
 24   D=D+WI(I,J)*Z(J)
      PO(I)=-D
 30   PONORM=PONORM+ABSF(D)
      ALPHA=-PONORM/10.
      DO 31 I=1,N
 31   PO(I)=PO(I)/ALPHA
      RETURN
 100  FORMAT(40H THE EQUIVALENT INITIAL STATE VECTOR IS, )
 101  FORMAT(1H ,E14.8,7E15.8)
 102  FORMAT (//37H THE EXACT INITIAL COSTATE VECTOR IS, )
 103  FORMAT (12H THE COST IS,E15.8,14HUNITS OF FUEL. )
      END
```

```
      SUBROUTINE START(QSUP)
C     CHOOSE AN INITIAL VALUE FOR ETA AND FOR ALPl.
      DIMENSION A(10,10),B(10),Q(10,101),PO(10),EMAT(10,10),AT(10,10),TH
     1ETA(10),ZETA(10),Z(10),        AIDENT(10,10),PERMU(10),WI(10,10),C
     12(10,10),W(10,10),X(10,101),C1(10),QSUP(10),EFETA(10)
      COMMON Q,PO,A,B,AT,Z,N,M,DELTA,DELSR,ETA,AIDENT,PERMU,L,WI,C2
     1,W,ALPHA,ALPH1,ICON,EFETA,ZNORM,ETNORM,X
      FUELF(X)=-DELTA/2.0*(TANHF((X+1.)*ETA)+TANHF((X-1.)*ETA))
      PNORM=ABSF(QSUP(1)*PO(1))
      DO 1I=2,N
      POFT=ABSF(QSUP(I)*PO(I))
 1    PNORM=MAX1F(PNORM,POFT)
      ETA=1./PNORM
C     NOW CHOOSE AN INITIAL VALUE FOR ALPH.  FIRST FIND EFETA(I).
      ARGA=0.
      ARGB=0.
      DO 12 I=1,N
      ARGA=ARGA+Q(I,1)*PO(I)
12    ARGB=ARGB+Q(I,M+1)*PO(I)
      DO 13 I=1,N
13    EFETA(I)=.5*(Q(I,1)*FUELF(ARGA)+Q(I,M+1)*FUELF(ARGB))
      DO 15 K=2,M
      ARG=0.
      DO 14 I=1,N
14    ARG=ARG+Q(I,K)*PO(I)
      DO 15 I=1,N
15    EFETA(I)=EFETA(I)+Q(I,K)*FUELF(ARG)
C     CALCULATE THE NORMS OF EFETA(I) AND Z(I)
      ETNORM=0.
      ZNORM=0.
      DO 16 I=1,N
      ETNORM=ETNORM+ABSF(EFETA(I))
16    ZNORM=ZNORM+ABSF(Z(I))
      ALPH=-ALPHA*ETNORM/ZNORM*(1.+ETA)
      IF(ALPHA+ALPH)8,7,7
7     ALPH1=0.
      RETURN
8     ALPH1=ALPHA+ALPH
      RETURN
      END



      SUBROUTINE ITER(EPS)
C     FIND THE NEW COSTATE VECTOR
      DIMENSION Q(10,101),PO(10),A(10,10),B(10),AT(10,10),Z(10),C1(10),C
     12(10,10),PL(10),AIDENT(10,10),PERMU(10),POST(10),WI(10,10)
     2,W(10,10),QSUP(10),EFETA(10),ZETA(10),EAT(10,10)
      COMMON Q,PO,A,B,AT,Z,N,M,DELTA,DELSR,ETA,AIDENT,PERMU,L,WI,C2
     1,W,ALPHA,ALPH1,ICON,EFETA,ZNORM,ETNORM,ZETA,EAT
C     COMPUTE THE CONTROL AND ITS FIRST DERIVATIVE
      FUELF(X)=-DELTA/2.0*(TANHF((X+1.)*ETA)+TANHF((X-1.)*ETA))
      FUDERF(X)=-DELTA/2.*ETA*(2.-(TANHF((X+1.)*ETA))**2-(TANHF((X-1.)*E
     1TA))**2)
      PRINT 127
      ICON=0
      DO 2 I=1,N
2     POST(I)=PO(I)
C     INITIALIZE FOR THE INTEGRALS OF ITERATION
 1    ARGB=0.
      ARGA=0.
      DET=1.
      DO 10 I=1,N
      ARGB=Q(I,M+1)*PO(I)     +ARGB
      ARGA=Q(I,1)*PO(I)+ARGA
```

```
 10   PL(I)=PO(I)
      PEN=.5*(ABSF(FUELF(ARGA))+ABSF(FUELF(ARGB)))
      DO 22 I=1,N
      C1(I)=.5*(Q(I,1)*FUELF(ARGA)+Q(I,M+1)*FUELF(ARGB))
      DO 11 J=1,N
      AIDENT(I,J)=0.
 11   C2(I,J)=.5*(Q(I,1)*Q(J,1)*FUDERF(ARGA)+Q(I,M+1)*Q(J,M+1)*FUDERF(AR
     1GB))
 22   AIDENT(I,I)=1.0
C     FIND THE INTEGRALS OF ITERATION
      DO 23 K=2,M
      ARG=0.
      DO 21 I=1,N
 21   ARG=ARG+Q(I,K)*PO(I)
      PEN=PEN+ABSF(FUELF(ARG))
      DO 23 I=1,N
      C1(I)=C1(I)+Q(I,K)*FUELF(ARG)
      DO 23 J=1,N
 23   C2(I,J)=C2(I,J)+Q(I,K)*Q(J,K)*FUDERF(ARG)
C     TAKE CARE OF THE CASE ALPH LESS THAN ALPHA.
      IF(ALPH1)46,50,50
 46   DO 47 I=1,N
      DO 47 J=1,N
      C1(I)=C1(I)+ALPH1*W(I,J)*PO(J)
 47   C2(I,J)=C2(I,J)+ALPH1*W(I,J)
C     NOW PUT THE WHOLE EXPRESSION TOGETHER
 50   L=XSIMEQN(10,N,N,C2,AIDENT,DET,PERMU)
      GO TO (54,55,32),L
 55   PRINT 106
 54   DO 25 I=1,N
 25   C1(I)=Z(I)+C1(I)
      DO 26 I=1,N
      DO 26 J=1,N
 26   PO(I)=PO(I)-C2(I,J)*C1(J)
C     CHECK FOR CONVERGENCE
      PON=0.
      ERROR=0.
      DO 30 I=1,N
      PON=PON+ABSF(PL(I))
 30   ERROR=ERROR+ABSF(PO(I)-PL(I))
      ERROR=ERROR/PON
      IF(ERROR-EPS)34,34,31
 31   PRINT 100,(PO(I),I=1,N)
100   FORMAT (40X,5E15.8)
      ICON=ICON+1
      IF(ICON-30)33,32,32
 33   IF(ERROR-30.)1,32,32
 32   ICON=0
      DO 40 I=1,N
 40   PO(I)=POST(I)
      RETURN
 34   ICON=1
      PRINT 103,PEN
 35   RETURN
 103  FORMAT (12H THE COST IS,E16.8,16H  UNITS OF FUEL. )
 106  FORMAT (87X33HTHE FIRST DERIVATIVE IS TOO FLAT. )
 127   FORMAT (60X56HNOW ITERATE ON THE COSTATE VECTOR, USING NEWTONS ME
     1THOD. )
      END
```

```
      SUBROUTINE CHGETA(ICOUNT,ALPT,KPETA)
C     TO CHANGE THE VALUE OF ETA AND OF ALPH
      DIMENSION A(10,10),B(10),Q(10,101),PO(10),EMAT(10,10),AT(10,10),TH
     1ETA(10),ZETA(10),Z(10),BIG(10),AIDENT(10,10),PERMU(10),WI(10,10),
     1C2(10,10),W(10,10),QSUP(10),EFETA(10),EAT(10,10)
      COMMON Q,PO,A,B,AT,Z,N,M,DELTA,DELSR,ETA,AIDENT,PERMU,L,WI,C2
     1,W,ALPHA,ALPH1,ICON,EFETA,ZNORM,ETNORM,ZETA,EAT
      FUELF(X)=-DELTA/2.0*(TANHF((X+1.)*ETA)+TANHF((X-1.)*ETA))
      IF(ICON)1,1,6
1     PRINT 10
10    FORMAT (38H  WE DID NOT CONVERGE.  NOW REDUCE ETA )
      IF(ICOUNT)3,4,4
3     ETA=ETA/2.
      ALPH1=.5*(ALPHA+ALPH1)
      RETURN
4     ICOUNT=ICOUNT+1
      IF(ALPH2)20,22,22
20    IF(ALPH1)23,21,22
21    ALPH=-ALPHA
      ALPINC=-ALPH2
      GO TO 23
22    KPETA=0
 23   IF(ICOUNT-KPETA)9,9,8
 8    ETA=ETA-.5**(ICOUNT-KPETA)*ETAINC
 9    ALPH=ALPH-.5**ICOUNT*ALPINC
      ALPH1=MIN1F(0.,ALPHA+ALPH)
      RETURN
 6    IF(ICOUNT)7,2,2
 7    ALPH=ALPH1-ALPHA
2     ICOUNT=0
      ETAINC=ETA*((60./FLOATF(N**3))**.25)
      ETA=ETA+ETAINC
C     NOW CHOOSE A NEW VALUE FOR ALPH.  FIRST FIND EFETA(I).
5     ARGA=0.
      ARGB=0.
      DO 12 I=1,N
      ARGA=ARGA+Q(I,1)*PO(I)
12    ARGB=ARGB+Q(I,M+1)*PO(I)
      DO 13 I=1,N
13    EFETA(I)=.5*(Q(I,1)*FUELF(ARGA)+Q(I,M+1)*FUELF(ARGB))
      DO 15 K=2,M
      ARG=0.
      DO 14 I=1,N
14    ARG=ARG+Q(I,K)*PO(I)
      DO 15 I=1,N
15    EFETA(I)=EFETA(I)+Q(I,K)*FUELF(ARG)
C     CALCULATE THE NORMS OF EFETA(I)
      ETNORM=0.
      DO 16 I=1,N
16    ETNORM=ETNORM+ABSF(EFETA(I))
      ALPINC=-ALPHA*ETNORM/ZNORM*(1.+ETA)*ALPT
      ALPH=ALPH+ALPINC
      ALPH2=ALPH1
      ALPH1=MIN1F(0.,ALPHA+ALPH)
      RETURN
      END
```

```
      SUBROUTINE SSTRAJ
C     TO COMPUTE THE CONTROL AND THE STATE SPACE TRAJECTORY.
      DIMENSION Q(10,101),PO(10),A(10,10),B(10),AT(10,10),Z(10),C1(10),C
     12(10,10),AIDENT(10,10),PERMU(10),WI(10,10),EMAT(10,10),EAT(10,10)
     2,W(10,10),QSUP(10),EFETA(10),E(10),D(10),X(10),ZETA(10)
      COMMON Q,PO,A,B,AT,Z,N,M,DELTA,DELSR,ETA,AIDENT,PERMU,L,WI,C2
     1,W,ALPHA,ALPH1,ICON,EFETA,ZNORM,ETNORM,ZETA,EAT
      FUELF(X)=-DELTA/2.0*(TANHF((X+1.)*ETA)+TANHF((X-1.)*ETA))
      ARG=0.
      DO 1 I=1,N
      X(I)=ZETA(I)
1     ARG=ARG+Q(I,1)*PO(I)
      PRINT 125
      PRINT 126
      PRINT 127,ARG,(X(I),I=1,N)
      MI=M+1
      DO 7 K=2,MI
      DO 2 I=1,N
2     X(I)=X(I)+.5*B(I)*FUELF(ARG)
      ARG=0.
      DO 3 I=1,N
      ARG=ARG+Q(I,K)*PO(I)
      PERMU(I)=0.
      DO 3 J=1,N
3     PERMU(I)=PERMU(I)+EAT(I,J)*X(J)
      DO 4 I=1,N
      X(I)=PERMU(I)
4     X(I)=X(I)+.5*B(I)*FUELF(ARG)
7     PRINT 127,ARG,(X(I),I=1,N)
      RETURN
125   FORMAT(//50X30HTHE STATE SPACE TRAJECTORY IS,/)
126   FORMAT(4X8H CONTROL,9X2HX1,13X2HX2,13X2HX3,13X2HX4,13X2HX5,13X2HX6,1
     1,13X2HX7)
127   FORMAT(1H ,E14.8,7E15.8)
      END
```

```
      SUBROUTINE CKCON(H)
      DIMENSION Q(10,101),PO(10),A(10,10),B(10),AT(10,10),Z(10),C1(10),C
     12(10,10),PL(10),AIDENT(10,10),PERMU(10),POST(10),WI(10,10),CO(10)
     2,W(10,10),QSUP(10),EFETA(10),CA(10),CB(10),C(10),ZETA(10),EAT(10,10)
     10)
      COMMON Q,PO,A,B,AT,Z,N,M,DELTA,DELSR,ETA,AIDENT,PERMU,L,WI,C2
     1,W,ALPHA,ALPH1,ICON,EFETA,ZNORM,ETNORM,ZETA,EAT
      FUELF(X)=-DELTA/2.0*(TANHF((X+1.)*ETA)+TANHF((X-1.)*ETA))
      FUDERF(X)=-DELTA/2.*ETA*(2.-(TANHF((X+1.)*ETA))**2-(TANHF((X-1.)*E
     1TA))**2)
      FSDERF(X)=DELTA*ETA**2*(TANHF((X+1.)*ETA)-TANHF((X+1.)*ETA)**3+TAN
     1HF((X-1.)*ETA)-TANHF((X-1.)*ETA)**3)
C     FIRST COMPUTE THE FIRST DERIVATIVE OPERATOR
      DET=1.
      ARGA=0.
      ARGB=0.
      DO 1 I=1,N
      ARGA=ARGA+Q(I,1)*PO(I)
1     ARGB=ARGB+Q(I,M+1)*PO(I)
      DO 4 I=1,N
      C1(I)=.5*(Q(I,1)*FUELF(ARGA)+Q(I,M+1)*FUELF(ARGB))
      DO 2 J=1,N
      C2(I,J)=.5*(Q(I,1)*Q(J,1)*FUDERF(ARGA)+Q(I,M+1)*Q(J,M+1)*FUDERF(AR
     1GB))
```

```
 2      AIDENT(I,J)=0.
 4      AIDENT(I,I)=1.
        DO 7 L=2,M
        ARG=0.
        DO 5 I=1,N
 5      ARG=ARG+Q(I,L)*PO(I)
        DO 7 I=1,N
        C1(I)=C1(I)+Q(I,L)*FUELF(ARG)
        DO 7 J=1,N
 7      C2(I,J)=C2(I,J)+Q(I,L)*Q(J,L)*FUDERF(ARG)
C       TAKE CARE OF THE USE ALPH1 NOT ZERO.
        IF(ALPH1)6,8,8
 6      DO 9 I=1,N
        DO 9 J=1,N
        C1(I)=C1(I)+ALPH1*W(I,J)*PO(J)
 9      C2(I,J)=C2(I,J)+ALPH1*W(I,J)
C       NOW GET THE INVERSE
 8      L=XSIMEQF(10,N,N,C2,AIDENT,DET,PERMU)
        GO TO (11,10,10),L
 10     PRINT 100
        H=0.
        RETURN
C       NOW FIND THE SECOND DERIVATIVE OPERATOR.
 11     ARGA=0.
        ARGB=0.
        DO 3 I=1,N
        ARGA=ARGA+Q(I,1)*PO(I)
 3      ARGB=ARGB+Q(I,M+1)*PO(I)
        DO 13 I=1,N
        CA(I)=0.
        CB(I)=0.
        DO 12 J=1,N
        CA(I)=CA(I)+C2(I,J)*Q(J,1)
 12     CB(I)=CB(I)+C2(I,J)*Q(J,M+1)
        DO 13 J=1,N
        DO 13 K=1,N
        CO(I)=.5*(ABSF(CA(I)*Q(J,1)*Q(K,1)*FSDERF(ARGA))+ABSF(CB(I)*Q(J,M+
       11)*Q(K,M+1)*FSDERF(ARGB)))
 13     C(I)=.5*(ABSF(CA(I)*Q(J,1)*Q(K,1))+ABSF(CB(I)*Q(J,M+1)*Q(K,M+1)))
        DO 15 L=2,M
        ARG=0.
        DO 20 I=1,N
 20     ARG=ARG+Q(I,L)*PO(I)
        DO 15 I=1,N
        CA(I)=0.
        DO 14 J=1,N
 14     CA(I)=CA(I)+C2(I,J)*Q(J,L)
        DO 15 J=1,N
        DO 15 K=1,N
        CO(I)=CO(I)+ABSF(CA(I)*Q(J,L)*Q(K,L)*FSDERF(ARG))
 15     C(I)=C(I)+ABSF(CA(I)*Q(J,L)*Q(K,L))
        ANORM=0.
        BNORM=0.
        CNORM=0.
C       NOW PUT TOGETHER THE OPERATOR EXPRESSIONS.
        DO 16 I=1,N
        C1(I)=C1(I)+Z(I)
 16     CA(I)=0.
        DO 18 I=1,N
        DO 17 J=1,N
 17     CA(I)=CA(I)+C2(I,J)*C1(J)
        ANORM=MAX1F(ANORM,ABSF(CA(I)))
        CNORM=MAX1F(CNORM,CO(I))
```

```
18    BNORM=MAX1F(BNORM,C(I))
      H=ANORM*BNORM*BIG(ETA)
      CNORM=CNORM*ANORM
      PRINT 111
      PRINT 110,ANORM,BNORM
      PRINT 112,CNORM
      RETURN
100   FORMAT(40X70H'THE FIRST DERIVATIVE IS TOO FLAT TO CHECK FOR CONVER
     1GENCE ACCURATELY.)
110   FORMAT(75X3E15.8)
111   FORMAT(75X37H  GAMMA*TO       GAMMA*T2           FN2)
112   FORMAT (80X24HTHE LOWER BOUND FOR H IS ,E15.8)
      END


      FUNCTION BIG(ETA)
      IF(ETA-.28)3,3,4
3     X=1./3.**.5-.62*ETA+2.*ETA**3
      GO TO 5
4     X=1./3.**.5*(1.-EXPF(-5.*ETA))
5     TE=TANHF(2.*ETA)
      DO 2I=1,6
      Y=(X+TE)/(1.+X*TE)
      X=X+.25*((1.-Y**2)*(1.-3.*Y**2)+(1.-X**2)*(1.-3.*X**2))/(Y*(2.-3.*
     1Y**2)*(1.-TE**2)/(1.+TE*ETA)**2+X*(2.-3.*X**2))
2     PRINT 110,X,Y
110   FORMAT (90X2E15.8)
      BIG=ETA**2*(X-X**3+Y-Y**3)
      PRINT 110,BIG
      RETURN
      END
```

NOMENCLATURE

$\underline{A}$	Plant matrix from the state equations (Chapter II)
$\underline{b}$	Control vector from the state equations (Chapter II)
$c, c_1, c_2 \ldots c_8, c_{\underline{\pi}_\ell}$	Constants
det (·)	The determinant of
dez (·)	The deadzone function of · (Chapter II)
$\underline{e}^{\underline{A}t}$	The exponential matrix of $\underline{A}t$ (Appendix A)
f	Orbital frequency of satellite (Chapter VI)
$\underline{f}(\underline{x}(t), \underline{u}(t), t)$	Nonlinear state equations (Chapter VII)
h	Guaranteed convergence parameter (Appendix B)
h_1	Estimated convergence parameter (Chapter II)
H	The Hamiltonian (Chapter II)
i, j, k, ℓ	Integer valued indices and constants
i	The $i^{\underline{th}}$ iteration of Newton's method, e.g., $\underline{\pi}^i$ (Chapter II)
k	The $k^{\underline{th}}$ approximate operator, e.g., $T_k(\underline{\pi})$ (Chapter II)
J(u)	The fuel functional (penalty function) (Chapter II)
M	Number of mesh points used in the computer program (Chapter IV)
n	Number of dimensions of the state space
$\underline{p}(t)$	The costate vector (Chapter II)
$\underline{q}(t) = \underline{e}^{-\underline{A}t}\underline{b}$	
R_n	n dimensional vector space (Appendix A)
$S_r(\underline{z})$	The Hypersphere of radius r about the point $\underline{z}$ (Appendix A)
t	Time
T	Final time--used for linear time-invariant Equations (Chapter II)
t_0, t_1	Initial and final times (Chapter VII)
T (·)	Exact operator (Chapter II)
$T_k(\cdot)$	$k^{\underline{th}}$ approximate operator (Chapter II)
TPBVP	Abbreviation for the Two Point Boundary Value Problem

NOMENCLATURE (Contd.)

u(t)	Control variable
$u^*(t)$	Fuel-optimal control (Chapter II)
$u_k(t)$	$k^{\underline{th}}$ approximate control (Chapter II)
$U_{(t_0, t_1]}$	Space of allowable control functions for $t \epsilon (t_0, t_1]$ (Appendix A)
$\underline{W}(T)$	Controlability matrix (Chapter II)
$\underline{x}(t)$	The state vector (Chapter II)
$\underline{y}, \underline{z}$	Dummy (vector) variables (Chapter III)
a_k	Amount of linear slope added to the control function in the $k^{\underline{th}}$ approximate operator $T_k(\underline{\pi})$ Chapter II)
Γ	Inverse of the first derivative operator (Chapter II)
Δ	Change in a quantity, e.g., $\Delta\eta$ is the change in η (Chapter IV)
$\delta, \epsilon, \epsilon_1, \epsilon_2$	Small constants
$\underline{\zeta}$	Equivalent initial state condition (Chapter IV)
η_k	A parameter of the approximate control function u_k (Chapter II)
$\underline{\theta}$	Terminal state boundary condition (Chapter II)
$\underline{\xi}$	Initial state boundary condition (Chapter II)
$\underline{\pi}$	Initial costate boundary condition (Chapter II)
$\pi = 3.14156$	
$\underline{\phi}(t, t_0)$	Fundamental matrix (Appendix A)
ω	Natural frequency radians/second
1(t)	Unit step function
$\|\cdot\|$	Norm (Appendix A)
$\equiv$ or $\stackrel{\Delta}{=}$	Defined to be
*	Optimal quantity, e.g., $\underline{x}^*(t)$
(1)	Derivative with respect to the argument, e.g, $u_k^{(1)}[\cdot]$ (Appendix A)
	Time derivative, e.g., $\dot{\underline{x}}(t)$

BIBLIOGRAPHY

1. Al'brekht, E. G., "On the Optimal Stabilization of Nonlinear Systems," Prikladnaya Matematica i Mekanika, Vol. 25, No. 5 1961.

2. Anselone, P. M., Editor, Nonlinear Integral Equations, Univ. of Wisconsin Press, 1964.

3. Antosiewicz, H. A., and Rheinboldt, W. C., "Numerical Analysis and Functional Analysis,"Chapter 14, Survey of Numerical Analysis, (J. Todd, editor), McGraw-Hill Book Co., New York, N.Y., 1962.

4. Athans, M., and Falb, P. L., Optimal Control, McGraw-Hill Book Co., New York, N.Y., 1966.

5. Athans, M., "The Status of Optimal Control Theory and Applications for Deterministic Systems," IEEE, PGAC, Vol. AC-11, No. 3, July, 1966, pp. 580-595.

6. Athans, M., "On the Uniqueness of the Extremal Controls for a Class of Minimum Fuel Problems," IEEE, PGAC, Vol. AC-11, No. 4, October, 1966, pp. 660-668.

7. Athans, M., and Canon, M. D., "On the Fuel-Optimal Singular Control of Nonlinear Second Order Systems," IEEE, PGAC, Vol. AC-9, No. 4, 1964, pp. 360-370.

8. Athans, M., et al, "Time-, Fuel-, and Energy-Optimal Control of Nonlinear Norm Invariant Systems," IEEE, PGAC, Vol. AC-8, No. 3, 1963, pp. 196-202.

9. Axelband, E. I., "An Approximation Technique for the Optimal Control of Linear Distributed Parameter Systems with Bounded Inputs," IEEE, PGAC, Vol. AC-11, No. 1, January, 1966, pp. 42-45.

10. Balakrishnan, A. V., and Neustadt, L. W., Editors, Computing Methods in Optimization Problems, Academic Press, New York, N.Y., 1964.

11. Barr, R. O., "Computation of Optimal Controls by Quadratic Programming on Convex Reachable Sets," Ph D Thesis, Univ. of Michigan, 1966.

12. Bartle, R. G., "Newton's Method in Banach Spaces," Proc. American Math. Society, Vol. 6, 1955, pp. 327-331.

13. Bass, R. W., "Optimal Feedback Control System Design by the Adjoint System," Technical Report 60-22A, Aeronica, Baltimore, Md., June 1960.

BIBLIOGRAPHY (Contd.)

14. Beckenbach, E. F., Editor, Modern Mathematics for the Engineer, Second Series, Chapter 1, "From Delta Functions to Distributions," by A. Erdelyi, McGraw-Hill Book Co., New York, New York, 1961.

15. Bellman, R., Editor, Mathematical Optimization Techniques, University of California Press, Berkeley, Calif., 1963.

16. Bellman, R., Kalaba, R., and Kotkin, "Some Numerical Results Using Quasilinearization for Nonlinear Two-Point Boundary Value Problems," Rand Report RM3113-PR, April, 1962.

17. Bellman, Kagiwada, and Kalaba, "Orbit Determination as a Multipoint B. V. Problem and Quasilinearization," Rand RM-3129-PR, May 1962.

18. Blum, E. K., Minimization of Functionals with Equality Constraints," United Aircraft Research Report C-110058-14, July, 1964.

19. Breakwell, J. V., Speyer, J. L., and Bryson, A. E., "Optimization and Control of Nonlinear Systems Using the Second Variation," S.I.A.M. Journal of Control, Series A, Vol. 1, No. 2, 1963, pp. 193-223.

20. Bryson, A. E., and Denham, W. F., "A Steepest-Ascent Method for Solving Optimum Programming Problems," J. Appl. Mech., Vol. 29, No. 2, 1962, pp. 247-257.

21. Chang, S. S. L., Synthesis of Optimal Control Systems, McGraw-Hill Book Co., 1961.

22. Craig, A. J., and Flügge-Lotz, I., "Investigation of Optimal Control with a Minimum-Fuel Consumption Criterion for a Fourth Order Plant with Two Control Inputs; Synthesis of an Efficient Suboptimal Control," J. A. C. C., preprints, June, 1964., pp. 207-221.

23. DeBra, D. B., "The Large Attitude Motions and Stability, Due to Gravity, of a Satellite with Passive Damping in an Orbit of Arbitrary Eccentricity about an Oblate Body," Stanford Univ., SUDAER, No. 126, Stanford, Calif., May, 1962.

24. Denham, W. F., "Steepest Ascent Solution of Optimal Programming Problems," Report BR 2393, Raytheon Co., Space and Info. Sys. Div., Bedford, Mass., April, 1963.

BIBLIOGRAPHY (Contd.)

25. Dieudonné, J., Foundations of Modern Analysis, Academic Press, New York and London, 1960.

26. Durbeck, R. C., "An Approximation Technique for Sub-optimal Control," IEEE, PGAC, Vol. AC-10, No. 2, April, 1965, pp. 144-150.

27. Eaton, J. H., "An Iterative Solution to Time-Optimal Control," J. Math. Anal. and Appl., Vol. 5, October, 1962, pp. 329-344.

28. Ehrlich, L., "Experience with Numerical Methods for a Boundary Value Problem," Technical Note N-N141, Space Technology Labs., Los Angeles, 1960.

29. Fancher, P. S., "Iterative Computation Procedures for an Optimum Control Problem," IEEE, PGAC, Vol. AC-10, No. 3, July, 1965, pp. 346-348.

30. Fox, L., The Numerical Solution of Two-Point Boundary Problems in Ordinary Differential Equations, Clarendon Press, Oxford, England, 1957.

31. Fröberg, C. E., Introduction to Numerical Analysis, Addison-Wesley Pub. Co., Reading, Massachusetts, 1965.

32. Gelfand, I. M., and Fomin, S. V., Calculus of Variations, translated by R. A. Silverman, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1963.

33. Gilbert, E. G., "An Iterative Procedure for Computing the Minimum of a Quadratic Form on a Convex Set," J. S.I.A.M. on Control, Vol. 4, No.1, 1966, pp. 61-80.

34. Gilbert, E. G., "The Application of Hybrid Computers to the Iterative Solution of Optimal Control Problems," pp. 261-284 of Computing Methods in Optimization Problems, (Balakrishnan and Neustadt, eds.), Academic Press, New York, 1964.

35. Goldstein, A. A., "Convex Programming and Optimal Control," J. S.I.A.M. on Control, Vol. 3, No. 1, 1965, pp. 142-146.

36. Gottlieb, R. G., "Rapid Convergence to Optimal Solutions Using a Min-H Strategy," J.A.C.C., 1966, pp. 167-176.

37. Halkin, H., "Liapounov's Theorem on the Range of a Vector Measure and Pontryagin's Maximum Principle," Arch. Rat. Mech. and Anal., Vol. 10, 1962, pp. 296-304.

BIBLIOGRAPHY (Contd.)

38. Haussler, R. L., "On the Suboptimal Design of Nonlinear Control Systems," Ph D Thesis, Purdue Univ., Lafayette, Ind., 1963.

39. Haussler, R. L., and Rekasius, Z. V., "Über die Suboptimale Regelung von Nichtlinearen Systemen," Regelungstechnik, July, 1964

40. Henrici, P., Elements of Numerical Analysis, Wiley, 1964.

41. Hestenes, M. R., Calculus of Variations and Optimal Control Theory, John Wiley and Sons, New York, August, 1966.

42. Hestenes, M. R., "Numerical Methods of Obtaining Solutions of Fixed End Point Problems in the Calculus of Variations," Rand RM-102, August, 1949.

43. Ho, Y. C., "A Successive Approximation Technique for Optimal Control Systems Subject to Input Saturation," Trans. ASME, J. of Basic Eng., Vol. 84, Series D, 1962, pp. 33-40.

44. Ho, Y. C. and Brentani, "On Computing Optimal Control with Inequality Constraints," J. SIAM on Control, Series A, Vol. 1, No. 3, 1963, p. 319.

45. Hurewicz, W., Lectures on Ordinary Differential Equations, The M.I.T. Press, Cambridge, Massachusetts, 1958.

46. John, F., Advanced Numerical Analysis, New York Univ. Institute of Mathematical Sciences, 1956.

47. Kalaba, R., "On Nonlinear Differential Equations, the Maximum Operation, and Monotone Convergence," Journal of Math. and Mech., Vol. 8, No. 4, July, 1959, pp. 519-574.

48. Kantorovich, L., "The Method of Successive Approximations for Functional Equations," Acta Math., Vol. 71, 1939, pp. 63-97.

49. Kantorovich, L. V. and Akilov, G. P., Functional Analysis in Normed Spaces, trans. by D. E. Brown, Edited by Dr. A. P. Robertson, Univ. of Glasgow, the Macmillan Co., New York, 1964. (A Pergamon book.)

50. Kantorovich, L. V. and Krylov, V. I., Approximate Methods of Higher Analysis, trans. by C. D. Benster, P. Noordhoff Ltd., Groningen, the Netherlands, 1958.

51. Kelley, H. J., "An Optimal Guidance Approximation Theory," IEEE, PGAC, Vol. AC-9, No. 4, October, 1964, pp. 375-380.

BIBLIOGRAPHY (Contd.)

52. Kelley, H. J., Kopp, R., and Moyer, G., "A Trajectory Optimization Technique Based upon the Theory of the Second Variation," A.I.A.A. Astrodynamics Conference, Yale Univ., New Haven, Conn., August, 1963.

53. Kelley, H. J., 'Method of Gradients," Chapter 6, Optimization Techniques, (Edited by G. Leitmann), Academic Press, New York, New York, 1962.

54. Kelley, H. J., "Gradient Theory of Optimal Flight Paths," A.R.S. Journal, Vol. 30, 1960, pp. 947-954.

55. Kipiniak, W., Dynamic Optimization and Control, The M.I.T. Press, Cambridge, Massachusetts, 1961.

56. Kleinman, D. L., "Suboptimal Design of Linear Regulator Systems Subject to Computer Storage Limitations," M.I.T. Ph D Thesis, February, 1967, also M.I.T., Report ESL-R-297.

57. Knapp, C. H., and Frost, P. A., "Determination of Optimum Control and Trajectories using the Maximum Principle in Association with a Gradient Technique," IEEE, PGAC, Vol. AC-10, No. 2, April, 1965, pp. 189-193.

58. Knudsen, H. K., "An Iterative Procedure for Computing Time-Optimal Controls," IEEE, PGAC, Vol. AC-9, No. 1, January, 1964, pp. 23-30.

59. Kolmogorov, A. N., and Fomin, S. V., Functional Analysis, Vol. 1, Trans. by L. F. Boron, Graylock Press, Rochester, New York, 1957.

60. Kopp, R. E., McGill, R., Moyer, H. G., and Pinkham, G., Several Trajectory Optimization Techniques; Part I, "Discussion" and Part II, "Application;" pp. 65-105 of "Computing Methods in Optimization Problems," (Balakrishnan and Neustadt, eds.), Academic Press, New York, 1964.

61. Lasalle, and Lefschetz, Nonlinear Differential Equations and Nonlinear Mechanics, Academic Press, New York, 1963.

62. Lee, E. B., "A Computational Technique for Optimal Systems," IEEE, PGAC, Vol. AC-10, No. 3, July, 1965, pp. 368-369 (corr.).

63. Leitmann, G., Editor, Topics in Optimization, Academic Press, New York, N. Y., 1967.

64. Leitmann, G., Editor, Optimization Techniques with Applications to Aerospace Systems, Academic Press, New York, N. Y., 1962.

BIBLIOGRAPHY (Contd.)

65. McGill, R., "Optimal Control, Inequality State Constraints, and the Generalized Newton-Raphson Algorithm," J. S.I.A.M. on Control, Vol. 3, No. 2, 1965, pp. 291-298.

66. McGill, R., and Kenneth, P., "Solution of Variational Problems by Means of a Generalized Newton-Raphson Operator," Grumman Res. Dept., Rpt. RE-176J, Bethpage, New York, May, 1964.

67. McGill, R. and Kenneth, P., A. Convergence Theorem on the Iterative Solution of Nonlinear Two-Point Boundary Value Systems," XIVth International Astronautical Congress, Paris, France, September 1963.

68. McReynolds, S. R., and Bryson, A. E., Jr., "A Successive Sweep Method for Solving Optimal Programming Problems," J.A.C.C., June, 1965, pp. 551-555.

69. Mercer, R. J., "Boundary Value Solutions of Trajectory Problems," Tech. Note PA-2399-01/2 Space Technology Labs., Los Angeles, 1960.

70. Merriam, C. W. III, Optimization Theory and The Design of Feedback Control Systems, McGraw-Hill Book Co., New York, N.Y., 1964.

71. Merriam, C. W., "An Algorithm for the Interative Solution of a Class of Two-Point Boundary Value Problems,"S.I.A.M. Journal of Control, Series A, Vol. 2, No. 1, 1964, pp. 1-10.

72. Neustadt, L. W., "The Existence of Optimal Controls in the Absence of Convexity Conditions," J. Math. Anal. Appl., Vol. 7, 1963, pp. 110-117.

73. Neustadt, L. W.,"On Synthesizing Optimal Controls," 2nd IFAC Congress, Basle, Switzerland, August, 1963.

74. Neustadt, L. W., "Synthesizing Time Optimal Control Systems," J. Math. Anal. and Appl., Vol. 1, December, 1960, pp. 484-493.

75. Newton, Sir Isaac, "Of Analysis by Equations of an Infinite Number of Terms," in The Mathematical Works of Isaac Newton, (D. T. Whiteside, assembler), Johnson Reprint Corp., New York, 1964.

76. Newton, Sir Isaac, "A Treatise of the Method of Fluxions and Infinite Series, with its Application to the Geometry of Curve Lines," appended to the Principia, California Press.

BIBLIOGRAPHY (Contd.)

77. Paiewonsky, B., "Optimal Control: A Review of Theory and Practice," AIAA Journal, Vol. 3, 1965, pp. 1985-2006.

78. Paiewonsky, B., "Time Optimal Control of Linear Systems with Bounded Control," in International Symposium on Nonlinear Differential Equations and Nonlinear Mechanics, Academic Press, New York, 1963.

79. Paiewonsky, B., Woodrow, P., Terkelsen, F., and McIntyre, J., "A Study of Synthesis Techniques for Optimal Controllers," Tech. Doc. Rpt. No. ASD-TDR-63-239, A.F.S.C., Wright Patterson A.F.B., Ohio, 1963.

80. Payne, J. A., "Computational Methods in Optimal Control Problems," UCLA Ph D Thesis, Los Angeles, Calif., 1965; also U. C. Dept. of Engr. Report No. 65-8; also A.F.S.C. Wright Patterson A.F.B., Ohio, Tech. Report AFFDL-TR-65-50.

81. Plant, J. B., "An Iterative Procedure for the Computation of Fixed-Time Fuel-Optimal Controls," IEEE, PGAC, Vol. AC-11, No. 4, October, 1966, pp. 652-659.

82. Plant, J. B., "An Iterative Procedure for the Computation of Optimal Controls," Massachusetts Institute of Technology Ph D Thesis, June, 1965.

83. Plant, J. B., and Athans, M., "An Iterative Procedure for the Computation of Optimal Controls," Paper 13D, 4th IFAC Congress, London, 1966.

84. Pontryagin, L. S., Boltyanskii, V. G., Gamkrelidze, R. V., and Mishchenko, E. F., The Mathematical Theory of Optimal Processes, translated by K. N. Trirogoff, John Wiley and Sons, New York, N. Y., 1962.

85. Rekasius, Z. V., Suboptimal Design of Intentionally Nonlinear Controllers," IEEE, PGAC, Vol. AC-9, No. 4, October, 1964, pp. 380-386.

86. Reisz, Frigyes and Sz-Nagy, Béla, Functional Analysis, Translated from 2nd French Edition by Leo F. Boron, Frederick Ungar Pub. Co., New York, 1955. (Trans. from Legons D'Analyse Fonctionelle.)

87. Rubio, J. E., "A Fixed-Point Method for a Minimum-Norm Control Problem," J. S.I.A.M. on Control, Series A, Vol. 4, No. 4, November, 1966.

88. Schwartz, L., Théorie des distributions, Vols. 1 and 2, Hermann and Cie., Paris, 1951, 1957.

BIBLIOGRAPHY (Contd.)

89. Sylvester, R. J. and Meyer, F., "Two-Point Boundary Problems by Quasilinearization," Report No. ATN-64 (S4855-60)-1, Aerospace Corporation, San Bernardino, California, 1964.

90. Tou, J. S., Modern Control Theory, McGraw-Hill Book Co., New York, N. Y., 1964.

91. Traub, J. F., Iterative Methods for the Solution of Equations, Prentice Hall Inc., Englewood Cliffs, New Jersey, 1964.

92. Tricomi, F. G., Integral Equations, Interscience Publishers, New York, N. Y., 1957.

93. Vainberg, M. M., Variational Methods for the Study of Nonlinear Operators, with a chapter on "Newton's Method," by L. V. Kantorovich and G. P. Akilov, translated by Amiel Feinstein, Holden-Day Inc., San Francisco, Calif., 1964.

94. Van Dine, C. P., "An Application of Newton's Method to the Finite Difference Solution of Nonlinear Boundary Value Systems," Report UAR-D37, United Aircraft Corp., March 11, 1965.

95. Van Trees, H. L., Synthesis of Optimum Nonlinear Control Systems, The M.I.T. Press, Cambridge, Massachusetts, 1962.

96. Witsenhausen, H. S., "Minimax Control of Uncertain Systems," Massachusetts Institute of Technology Ph D Thesis, May, 1966, also E.S.L. Report ESL-R-269.

97. Witsenhausen, H. S., "Some Iterative Methods Using Partial Order for Solution of Nonlinear Boundary Value Problems," Technical Note 1965-18, Lincoln Laboratory, Lexington, Mass., May, 1965.

98. Zadeh, L. A., and Desoer, C. A., Linear System Theory, the State Space Approach, McGraw-Hill Book Co., Inc., New York, 1963.

99. Bellman, R.E., and Dreyfus, S. E., Applied Dynamic Programming, Princeton Univ. Press, Princeton, N. J., 1962.

100. Pshenichniy, B. N., "Linear Optimal Control Problems," J. S.I.A.M. Control, Vol. 4, No. 4, 1966, pp. 577-593.

BIBLIOGRAPHY (Contd.)

101. Paynter, H. M., Analysis and Design of Engineering Systems, The M.I.T. Press, Cambridge, Mass., 1960.

102. Denn, M.M., "Convergence of a Method of Successive Approximations in the Theory of Optimal Processes," Ind. and Eng. Chem. Fundamentals, Vol. 4, No. 2, May, 1965. p. 231.

103. Kurihara, H., "Optimal Control of Stirred-Tank Chemical Reactors," M.I.T. PhD thesis and also Electronic Systems Laboratory Report ESL-R-267, Massachusetts Institute of Technology, Cambridge, Mass., 1966.